NEU!

Das Magazin für den Home-Computer-Besitzer

computronic

6/1. Jahrgang Juni '84

DM 5,50 SFR 5,50 ÖS 45

Spielprogramme

Anwendungsprogramme

für

ZX-Spectrum

Defender
Lui der Wurm
Alternativer Zeichensatz

Apple II

Spape tables
Super Datei
Snake

Dragon 32/64

Wargames

Atari

Tomstone-City
Schneevogel

VC-64

Autostart
Roddpainter
Data-Generator
Bestellung

VC-20

Bestellschein
Glücky

ZX-81

ZX-Draw
Mooncrash

TI-99

Noah-2099
Jack the Digger

mit Forth-Kurs

Teil 3

Jetzt 80 Seiten!

Autoren gesucht!

Werbe Aktion
Billige Kleinanzeigen
Seite 78

Möchten Sie als Software-Autor für den TRONIC-Verlag tätig werden?

Wir würden uns freuen, in Ihnen einen Ansprechpartner für eine künftige intensive Zusammenarbeit zu finden. Der TRONIC-Verlag vereinbart mit seinen künftigen Software-Autoren pro veröffentlichte Seite (einschl. Programmbeschreibung) ein Honorar von **DM 120,—.**

Dieses Entgelt wird fällig, wenn die Redaktion des Verlages sich für eine Veröffentlichung entscheidet. Die Auszahlung erfolgt also nicht erst nach Veröffentlichung in einer unserer Ausgaben, sondern wesentlich früher.

Der Verlag wird vom Autor berechtigt, seine Manuskripte (Programme) zur Darstellung im Heft heranzuziehen und abzudrucken.

Einzusenden sind:

- Programmbeschreibung und Programmflußplan
- bespielte Cassette oder Diskette
- Listing

mit Copyright.

- Freiumschlag



Haben Sie Interesse? Haben Sie noch Fragen?

Setzen Sie sich telefonisch mit unserer Redaktion

in Verbindung!

TRONIC-VERLAG

DIE REDAKTION

Programmieren in Forth

Definition neuer Befehls-Worte Teil 3

Nachdem wir uns im letzten Teil mit den Arithmetik-Operationen beschäfitgt haben, wollen wir uns nun mit der Definition neuer Befehle befassen. Die bislang vorgestellten FORTH-Befehle sind bis jetzt ausschließlich interaktiv und interpretativ aufgerufen worden. Das bedeutet, daß ein Wort (z. B. DUP oder +) mit der Eingabe seines Namens in die Tastatur aufgerufen wird, indem der äußere Interpreter dann das Wörterbuch (Dictionary) nach diesem Wort absucht, es lokalisiert und ausführt. Diese Art der Befehls-Abarbeitung wird im Gegensatz zu Basic nur bei der direkten Eingabe von Befehlen verwendet. Wir haben nun bereits so viele Befehle kennengelernt, daß wir daran gehen können, kleinere Programme damit zu erstellen. Wie schon im ersten Teil erwähnt, ist das gleichbedeutend damit, die zu anfangs erwähnten Befehle zu neuen Befehlen zusammenzusetzen.

Wollen wir einen Befehl definieren, so müssen wir als erstes mit dem Befehl »:«, den FORTH-COMPILER einschalten. Als nächstes wird die Bezeichnung und danach die eigentliche Definition des neuen Befehles eingegeben. Mit dem Befehl »;« wird der FORTH-COMPILER wieder ausgeschaltet und damit die Definition abgeschlossen.

Wir wollen nun einen Befehl definieren, welcher eine Zahl mit sich selbst multipliziert und das Ergebnis auf den Stack legt. Als Bezeichnung wählen wir »QUADRO«.

```
: QUATRO DUP * ; ⟨RETURN⟩ OK
```

Wie zu erkennen ist, besteht die eigentliche Definition aus einer unbegrenzten Zahl von Befehlen. Der Befehl DUP dupliziert die oberste Zahl des Stack und der nachfolgende »*« Befehl bildet, durch die Multiplikation, die Quadratzahl.

In den nachfolgenden Beispielen können wir den neuen Befehl testen:

```
 2 QUATRO · ⟨RETURN⟩ 4 OK
 5 QUATRO · ⟨RETURN⟩ 25 OK
10 QUATRO · ⟨RETURN⟩ 100 OK
```

Durch den Befehl VLIST können wir das gesamte WÖRTERBUCH (BEFEHL's-Liste) auf Drucker oder Monitor ausgeben lassen. Der neue Befehl QUATRO steht nun am Anfang dieser Liste.

Bei der Defination von neuen Befehlen muß darauf geachtet werden, daß die Befehlsbezeichnung keine 31 Zeichen übersteigt. Es ist nicht unbedingt notwendig und auch nicht immer möglich, daß die Defination in eine einzige Zeile geschrieben wird. Als Beispiel wollen wir einen Befehl definieren, welcher die Kubikzahl eines Operanten errechnet:

```
: KUBIK DUP DUP * * ; ⟨RETURN⟩ OK
```

Wir können das gleiche auch wie folgt eingeben:

```
: KUBIK       ⟨RETURN⟩
    DUP DUP   ⟨RETURN⟩
    * *       ⟨RETURN⟩
    ;         ⟨RETURN⟩ OK
```

Bei der direkten Eingabe von Befehlen ist die Schreibweise kaum von Bedeutung, da die Struktur der Eingabe nicht gespeichert wird. Ganz anders ist dies bei der Programmierung mit dem Texteditor. Hier sollte man unbedingt auf Strukturierung und gute Dokumentation achten, um auch später noch Erweiterungen oder Änderungen vornehmen zu können. Wir werden uns später noch genauer mit diesem Thema auseinander setzen.

Nun wollen wir auch unseren KUBIK-Befehl testen:

```
 2 KUBIK . ⟨RETURN⟩ 8 OK
 5 KUBIK . ⟨RETURN⟩125 OK
10 KUBIK . ⟨RETURN⟩1000 OK
```

Werden in einem Programm oft die gleichen Befehls-Folgen verwendet, so ist es sinnvoll diese zu einem Befehl zusammenzufassen:

```
: 1+ 1+ ; ⟨RETURN⟩ OK
: 1- 1- ; ⟨RETURN⟩
: 2+ 2+ ; ⟨RETURN⟩
```

Einige dieser einfachen Befehls-Folgen sind im FORTH-Standart schon enthalten und wurden im letzten T eil schon vorgestellt. Es ist nun sehr leicht, die fehlenden Kombinationen zu definieren. Wird in FORTH ein Befehl zweimal definiert, so gilt immer die letzte Defination.

FORTH befehle können auch zur Umsetzung dienen. Bei den meisten farbigen Computern dient ein Zahlencode zur Bestimmung einer Farbe.

```
Farbcode des Commodore 64:  0 = Schwarz
                            1 = Weiß
                            2 = Rot
                            3 = Türkis
                            usw.
```

Wollen wir statt des umständlichen Codes einfach die Farbe eingeben, können wir dies durch die folgenden Definationen erreichen:

```
: SCHWARZ 0 ;  ⟨RETURN⟩
: WEISS   1 ;  ⟨RETURN⟩
: ROT     2 ;  ⟨RETURN⟩
: TÜRKIS  3   ⟨RETURN⟩
usw.
```

Die Funktion ist sehr einfach: Nach Eingabe der Farbbezeichnung legt der entsprechende Befehl den dazugehörigen Wert auf den Stack.

Es ist auch möglich mit den Farben zu rechnen:

```
WEISS ROT + .  ⟨RETURN⟩ 3 OK
```

Mit Hilfe eines PRINT Befehls können wir auch leicht eine Vokabel-Übersetzung realisieren. Der PRINT Befehl besitzt in FORTH folgende Syntax: ."text".

Elementare String-Befehle

Leider existiert in FORTH nur eine kleiner Befehlssatz zu Stringverarbeitung. Der FORTH-Kern enthält jedoch einige Worte für einfache Ein- und Ausgaben von Strings (Texten), die für viele Applikationen völlig ausreichen dürften.

CR

Dieser Befehl bewirkt auf dem Ausgabegerät (normal Monitor) Zeilenvorschub und Wagenrücklauf.

12 13 14 . . CR . (RETURN) 14 13
12 OK

. "TRONIC· CR . „VERLAG" (RETURN) TRONIC
Verlag OK

SPACE

Ein SPACE-Zeichen wird ausgegeben

. „APRIL" SPACE . „MAI" (RETURN)
APRIL MAI OK

Spaces (n1 → empty)

Eine bestimmte Anzahl von SPACE-Zeichen wird ausgegeben.

. „APRIL" 4 SPACE . „MAI" (RETURN) APRIL MAI OK

EMIT (c → empty)

Der ASCII-Charakter c wird ausgegeben. In Basic lautet dieser Befehl CHR$ (x) .

65 EMIT (RETURN) A OK
66 EMIT (RETURN) B OK

EXPECT (adr n → empty)

Dieser Befehl entspricht in etwa dem INPUT des Basic-Interpreters. Ein String mit ‚n'-Zeichen wird von der Tastatur erwartet und ab Adresse ‚adr' abgespeichert. Ein RETURN kann diese Prozedur beenden, auch wenn die Anzahl ‚n' noch nicht erreicht sein sollte.

10000 11 EXPECT (RETURN)
COMPUTRONIC (RETURN) OK

TYPE (adr n → empty)

Dieser Befehl gibt eine Zeichenkette, die ab Adresse ‚adr' gespeichert und ‚n'-Zeichen lang ist, auf dem Ausgabegerät aus.

10000 11 TYPE (RETURN) COMPUTRONIC OK

COUNT (adr1 → adr2 n)

Dieser Befehl erwartet eine Adresse ‚adr 1' auf dem Stack und ermittelt hieraus die Startadresse und die Zeichenzahl des Strings. Dies sind die Angaben, die von Type erwartet werden.



Die Anwendung dieses Wortes erfolgt also üblicherweise in der Form ‚adr1 COUNT TYPE', wobei als ‚adr1' die Adresse des COUNT-Bytes (Zeichenanzahl) übergeben wird. Der eigentliche String muß dem COUNT-Byte folgen.

KEY (→ c)

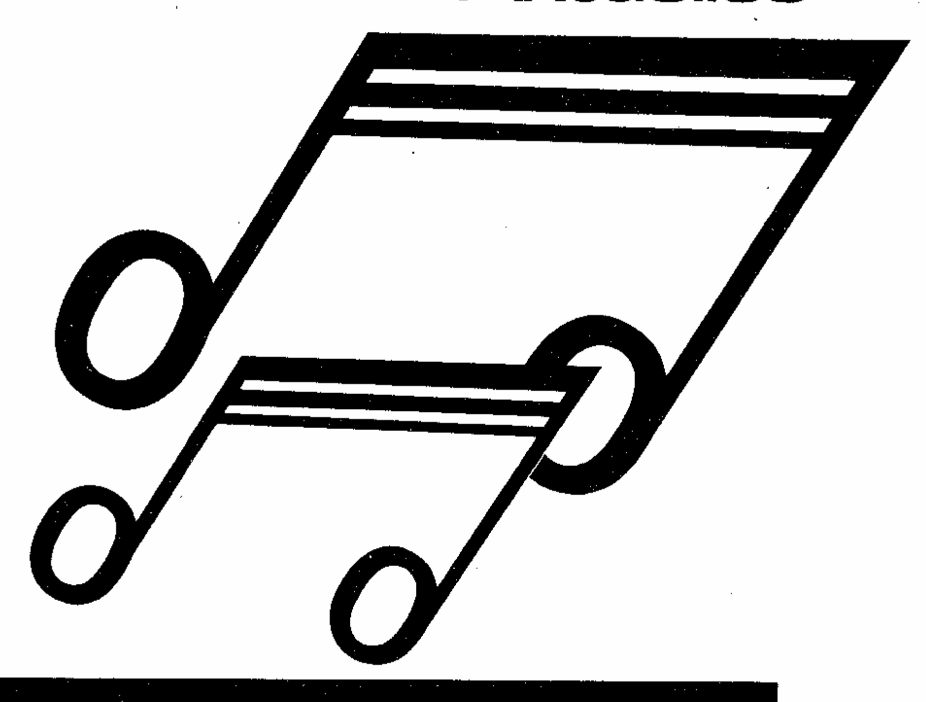
Dieser Befehl wartet auf eine beliebige Tastatureingabe und übergibt den entsprechenden ASCII-Code auf den Stack. Bei einigen Forth-Versionen ist die RETURN-Taste zur Quittierung erforderlich.

KEY . (RETURN)
A
65 OK

KEY KEY . . (RETURN)
A B
66 65 OK

KEY EMIT (RETURN)
D
D OK





Klang aus dem Chip

Computer können nicht nur rechnen, zählen und sortieren. Die elektronischen Heinzelmännchen können weit mehr. Musik machen zum Beispiel. Heimcomputer spielen wie Klaviere, wie Saxophone oder wie Schlagzeuge. Sie spielen ein-und mehrstimmig – je nach Wunsch

Keine echte Hitparade kommt ohne sie aus: Elektronische Klangeffekte liebte schon Rock-Vater Elvis, und seine Nachfahren – von den Beatles bis zu Nina Hagen – feilten eifrig an der Musik aus Stromimpulsen. Auch so ungewöhnliche Töne wie etwa das Geheul startender Raumschiffe in „Star Wars" und das Zischen von Science-fiction-Strahlenwaffen stammen aus dem Wunderland der Elektronik.

Aufwendig und teuer ist die Erzeugung solchen Sounds in den Tonstudios früherer Tage gewesen. Die Mischpult-Regisseure fürchten nun die Konkurrenz des Home-Computers. Der macht's entschieden billiger.

Jeder, der mit dem Keyboard eines derartigen Gerätes umgehen kann, ist sein eigener Toningenieur. Auf Tastendruck lassen sich den Chips die unmöglichsten Geräusche entlocken. Aber auch den Ohren genehme Töne und ganze Musikstückchen rieseln sanft aus dem Mikrocomputer.

„Eine irre Sache", findet Ernst Huber. Der 17 Jahre alte Schüler aus München hat seinen Commodore 64 in ein piepsendes, summendes und klingendes Musikinstrument verwandelt. Per Knopfdruck wird der „Computer des Jahres 1983" plötzlich zum Synthesizer.

Das kann übrigens jeder Mikrorechner. Nur der C 64 macht's besonders schön. In einem speziellen Bauteil – dem „Sound Interface Device (SID)" – sind die Klangelemente eingepaßt. Zu dem Chip gehören drei Tongeneratoren, mischbare Filter und sogenannte Ringmodulatoren als technischer Zusatz.

Damit läßt sich schon ein ganz erhebliches Spektakel anrichten. Donner, Motorgedröhn, Schüsse und Detonationen von Sprengstoff hören sich aber erst dann so richtig

echt an, wenn eine Stereo-Anlage mit dem Computer verbunden wird. Mit einem künstlichen Donner hat Ernst auf diese Weise seine große Schwester schon mal in heillosen Schrecken versetzt. Aber auch weniger unfriedliche Geräusche können mit der Tonerzeugung des C 64 hervorgebracht werden.

Das Programm „Synthimat 64" wandelt für knapp hundert Mark die Rechner-Tastatur in ein Klavier um. Zur Hardware-Ausrüstung sind neben dem Computer ein Diskettenlaufwerk und ein Farbfernseher nötig.

Auf eine Diskette passen 256 Soundprogramme, dazu neun vollständige Lieder, die sich jeweils mit den Sounds kombinieren lassen. So sind Begleitung und Melodiestimme unabhängig voneinander zu spielen. Das wird auch auf dem Bildschirm abgebildet. Zwei verschiedene Tastaturen tauchen auf dem Monitor auf.

Das Keyboard des Homecomputers ist freilich nicht für Pianistenhände geschaffen. Martha Argerich würde es bestimmt als Zumutung von sich weisen, auf schlichten Schreibmaschinentasten ein Konzert zu geben. Profimusiker werden wohl kaum auf ein echtes Klavier verzichten wollen. Doch Hobby-Spieler und Einsteiger in die Welt des elektronischen Klangs können sich über ihren Spar-Synthesizer freuen.

Und die Funktionen sind vielfältig, die „Synthimat" der Elektronik entlockt. Beispielsweise kann eine Melodie als Begleitmusik zu einem Lied gewählt werden. Auf Druck einer bestimmten Tastenfolge spielt das Programm die Tonleiter rauf und runter. Oder schwebende Töne aus den Oszillatoren gaukeln eine Streicherbegleitung vor. Auch Akkorde klingen aus dem zum Musikinstrument erkorenen Computer.

Jede Taste, die berührt wird, wird auf dem Keyboard-Schema des Bildschirms angezeigt – wie bei einem Klavier, das von Geisterhand spielt.

Besonders einfach ist es, den ZX Spectrum als tönendes Gerät, das die Tonleiter auswendig weiß, einzusetzen. Dazu wird die „BEEP"-Taste benutzt. Die Noten müssen zuvor in Zahlen codiert werden. Dann läuft ein selbst programmiertes Musikstück ab. So bedeutet etwa 1,0 die Tonlage eines C. Die Eins legt die Länge des Tones und die zweite Zahl die Höhe fest. Für eine Sekunde lang klingt ein C auf. Aber erst mit einem Verstärker sind die Musikstücke in angenehmer Lautstärke vernehmbar.

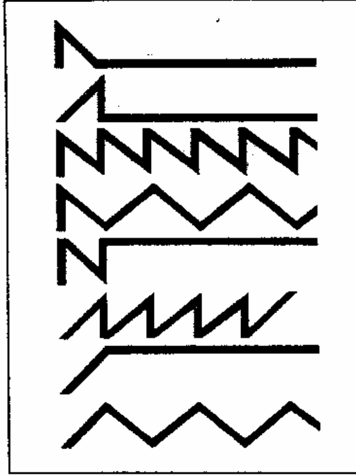
Mit einem maßgeschneiderten Programm tut der Spectrum auch mal so, als sei er ein Klavier.

Der lockere Umgang mit Musik fällt kaum einem Computer schwer.

Aber wie kommt es dazu, daß diese elektronischen Apparate überhaupt zu Komponisten und Solisten werden können? Die Elektrotechnik macht sich zunutze, daß jeder Ton – egal ob ein musikalischer oder sonst einer – aus Frequenzen besteht. Für das Ohr des Menschen ist ein Schwingungsbereich von 20 bis 15 000 Hertz hörbar. Der Computer produziert einfach Frequenzen. Aber noch andere Dinge kennzeichnen Töne. Wenn zum Beispiel eine Gitarrensaite angeschlagen wird, entsteht ein Ton, der erst laut und dann immer leiser wird. Die Anstiegszeit und die Abklingzeit beschreiben den charakteristischen Verlauf einer „Hüllkurve", die sozusagen das Profil eines Tones umreißt.

Diese Eigenheiten eines Klangs können mit Filter und Verstärker nachgeahmt werden. Im Computer sind Hüllkurvengeneratoren eingebaut, die diesen Vorgang steuern. Außerdem produziert der Computer Töne, deren Schwingungskurven wie Rechtecke, Sägezähne und Dreiecke aussehen. Sie klingen scharf, rauh oder weich. Und der Computer mischt sie. Erst ihre Mischung ergibt einen originalgetreuen Ton. Ein Synthesizer läßt programmgesteuert neben dem Grundton die entsprechenden Obertöne, die den Charakter eines bestimmten Klangs ausmachen, im richtigen Moment und mit der richtigen Stärke los. So ist es theoretisch möglich, alle existierenden Töne nachzumachen – auch mit einem Homecomputer. Bei Beethovens Neunte muß er allerdings passen. *Michael Dörfler*

Profil der Töne



Erst auf einem Oszillographen werden die Hüllkurven der Töne, die von einem Computer erzeugt werden, sichtbar. Sie sind für die Klangfarbe der Töne verantwortlich. Je nach Wahl der Hüllkurve klingt der Ton scharf, rauh oder weich

Beispiel für ein Musik-Programm

Programm	Erläuterung
10 SI = 54272: FL = SI: FH = SI+1: W = SI+4: A = SI+5: H = SI+6: L = SI+24	1. Definition der Registeradressen
20 POKE L, 15	2. Volle Lautstärke
30 POKE A, 16+9	3. Anschlag
40 POKER H, 4* 16+4	4. Halten und Ausklingen
50 POKE FH, 29: POKE FL, 69	5. Hi und Lo Byte der Frequenz, hier für den Kammerton A. Für andere Töne sind die beiden Werte aus der Tabelle im Anhang P zu entnehmen.
60 POKE W, 17	6. Wellenform. Muß immer als letztes eingestellt werden, da das niedrigste Bit in diesem Register den Tongenerator ein- bzw. ausschaltet.
70 FORT = 1T05000:NEXT	7. Schleife zur Einstellung der Tondauer.
80 POKE W, 0:POKE A, 0: POKE H, 0	8. Ausschalten von Wellenform- und Hüllkurveneinstellungen.

Mit freundlicher Genehmigung : Magazin "Hobby - Technik"

ZX Spectrum

» Defender «

Vernichten Sie die Invasoren

Für ZX-Spectrum 48 K

Sie sind mit Ihrem Raumschiff im Weltraum stationiert um eine Invasion der „Gologs" abzuwehren. Die Gologinvasoren greifen mit drei verschiedenen Raumschifftypen an. Mit ihren „Jägern", mit „Raketen" und mit „Lander".
Während der Spielphase stehen Ihnen 3 Raumschiffe zur Verfügung. Werden Ihre Schiffe von den Gologs gerammt, oder erreicht ein Landerfahrzeug der Invasoren den linken Bildschirmrand, haben Sie verloren. Für jede Rakete oder jeden Jäger der Gologs, die den Bildschirmrand erreichen, gibt es Punktabzüge.
Neu gestartet werden kann mit einer beliebigen Taste!

```
   1 REM DEFENDER
   2 LET hsc=0
   3 GO TO 9000
  10 BORDER 0: PAPER 0: BRIGHT 1: INK 7: CLS
  15 LET sc=0: LET c=11: LET schiff=3
  20 PRINT #0;AT 1,0;"SCHIFFE=";schiff;AT 1,12;"HSC=";hsc
  60 LET x=INT (RND*3)+1
  70 IF x=1 THEN  GO TO 100
  80 IF x=2 THEN  GO TO 200
  90 IF x=3 THEN  GO TO 300
 100 LET a=INT (RND*20): LET b=29
 110 LET a1=a: LET b1=b: LET x=INT (RND*2)+1
 120 LET b=b-1: IF b=0 THEN  GO TO 2000
 130 LET a=a+(x=1 AND a<20)-(x=2 AND a>1)
 140 PRINT AT a1,b1;"  ";AT a,b; INK 5; BRIGHT 1;"[illegible]"
 150 IF INT (RND*5)+1=3 THEN  GO SUB 800
 160 GO SUB 500
 170 GO TO 110
 200 LET a=INT (RND*21): LET b=29
 210 LET a1=a: LET b1=b
 215 LET b=b-2
 218 IF b<=1 AND a=c THEN  LET po=c: LET pe=0: GO SUB 1000: LET schiff=s
chiff-1: PRINT #0;AT 1,0;"SCHIFFE=";schiff: IF schiff=0 THEN  GO TO 2000

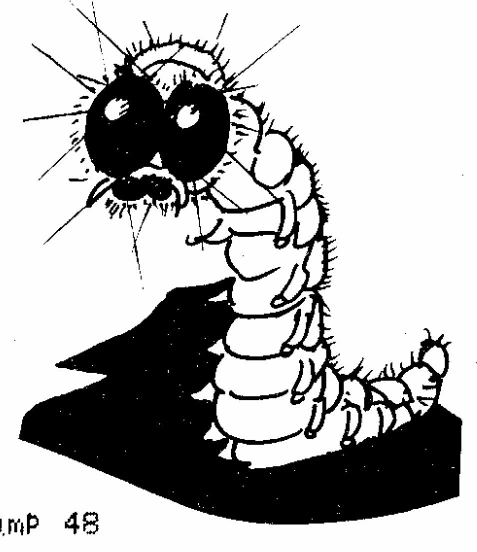
 220 IF b<=1 THEN  LET sc=sc-8: PRINT AT a,b;"  ";AT a1,b1;"   ": GO TO
60
 230 PRINT AT a1,b1;"  ";AT a,b; INK 2; BRIGHT 1;"[illegible]"
 240 IF INT (RND*3)+1=2 THEN  GO SUB 800
 250 GO SUB 500
 260 GO TO 210
 300 LET a=INT (RND*21): LET b=29
 310 LET j=INT (RND*20)
 320 IF a>j THEN  LET x=-1
 330 IF a<j THEN  LET x=1
 335 IF a=j THEN  GO TO 310
 340 LET a1=a: LET b1=b
 345 LET b=b-1
```

```
 350 IF b=1 AND a=c THEN  LET po=c: LET pe=0: GO SUB 1000: LET schiff=sc
hiff-1: PRINT #0;AT 1,0;"SCHIFFE=";schiff: IF schiff=0 THEN  GO TO 2000
 352 IF b=0 THEN  LET sc=sc-5: PRINT AT a,b;"  ";AT a1,b1;"   ": GO TO 6
0
 355 LET a=a+x
 358 IF a=i THEN  PRINT AT a1,b1;"  ": GO TO 310
 360 PRINT AT a1,b1;"  ";AT a,b; INK 3; BRIGHT 1;"[illegible]"
 370 IF INT (RND*8)+1=4 THEN  GO SUB 800
 380 GO SUB 500
 390 GO TO 340
 500 LET c1=c
 510 LET c=c+(INKEY$="z" AND c<21)-(INKEY$="a" AND c>1)
 520 IF INKEY$="m" THEN  GO SUB 600
 530 PRINT AT c1,0;"  ";AT c,0; INK 7; BRIGHT 1;"[illegible]"
 540 RETURN
 600 PLOT 17,175-(8*c)-5: DRAW  INK 6; BRIGHT 1;(8*b),0: BEEP .003,10: B
EEP .003,15
 610 PLOT  OVER 1;17,175-(8*c)-5: DRAW  OVER 1;(8*b),0: BEEP .003,10
 620 IF c=a THEN  LET po=a: LET pe=b: GO SUB 1000: LET sc=sc+b: PRINT #0
 AT 1,20;"SCORE=";sc;"  ": GO TO 60
 630 RETURN
 800 PLOT (8*b),175-(8*a)-5
 810 DRAW  INK 2; BRIGHT 1;-(8*b),0: BEEP .005,10
 820 PLOT  OVER 1;(8*b),175-(8*a)-5
 830 DRAW  OVER 1;-(8*b),0: BEEP .005,10
 840 IF a=c THEN  LET po=c: LET pe=0: GO SUB 1000: LET schiff=schiff-1:
PRINT #0;AT 1,0;"SCHIFFE=";schiff: IF schiff=0 THEN  GO TO 2000
 850 RETURN
1000 PRINT AT po,pe; INK 7; BRIGHT 1;"×": BEEP .03,-4
1010 FOR l=2 TO 7: PRINT AT po,pe; INK l; BRIGHT 1;"[illegible]": NEXT l
1020 PRINT AT po,pe; INK 2; BRIGHT 1;"[illegible]":BEEP .005,10: BEEP .01,1
1025 FOR l=1 TO 7: NEXT l
1030 FOR i=1 TO 2: FOR l=2 TO 7
1040 PRINT AT po,pe; INK l; BRIGHT 1;"[illegible]";AT po+1,pe; INK l; BRIGHT 1;"[illegible]
 ": BEEP .002,20
1050 NEXT l: NEXT i
1060 PRINT AT po+1,pe;"  ": FOR l=7 TO 2 STEP -1: PRINT AT po,pe; INK l;
"[illegible]" NEXT l
1070 BEEP .005,-10: FOR l=1 TO 5: PRINT AT po,pe; INK 6;"×": NEXT l
1080 PRINT AT po,pe;"  ": RETURN
2000 FOR i=1 TO 20: BEEP .01,10: NEXT i
2010 CLS : IF schiff=0 THEN  PRINT AT 11,0;"Schade alle Jaeger sind
           Zerstoehrt": PRINT "Du hast ";sc;" Punkte erreicht": GO TO
2100
2020 PRINT AT 11,0;"Die Invasoren sind Gelandet": PRINT "Du hast ";sc;"
Punkte erreicht"
2100 IF sc>hsc THEN  LET hsc=sc
2110 PRINT "Neues Spiel?  Druecke j oder n"
2120 IF INKEY$="" THEN  GO TO 2120
2130 IF INKEY$="j" THEN  CLS : GO TO 9470
2140 CLS : LET c$="Hat mich Gefreut Herr Kapitaen. Bis zum Naechstenmal
                    TSCUESS"
2145 PRINT AT 10,0;
2150 FOR i=1 TO LEN c$
2160 PRINT c$(i);: BEEP .005,-30+i: NEXT i
2170 STOP
9000 FOR i=1 TO 15: READ a$: FOR l=0 TO 7: READ a: POKE USR a$+l,a: NEXT
 l: NEXT i
9100 DATA "a",0,64,96,97,BIN 10010111,127,255,0
9110 DATA "b",0,0,96,BIN 11111000,255,12,BIN 11111000,0
```

```
9120 DATA "c",33,BIN 00010011,BIN 00001111,BIN 00001010,255,BIN 11111010
,BIN 00001111,3
9130 DATA "d",132,BIN 11001000,BIN 11110000,BIN 01010000,255,BIN 0101111
1,BIN 11110000,BIN 11000000
9140 DATA "e",8,24,BIN 01111100,255,255,BIN 01111100,24,8
9150 DATA "f",24,56,BIN 01001010,252,252,BIN 01001010,56,24
9160 DATA "g",0,0,4,BIN 00011111,BIN 00110000,255,8,0
9170 DATA "h",3,7,9,254,126,252,16,0
9180 DATA "i",129,0,36,152,25,36,0,129
9190 DATA "j",32,BIN 00110010,BIN 01111011,BIN 01101111,BIN 00001111,BIN
 00001101,BIN 00111101,BIN 01010000
9200 DATA "k",16,22,BIN 01111100,255,BIN 11110110,BIN 11100110,BIN 11001
110,64
9210 DATA "l",BIN 11000000,BIN 01101000,BIN 00100100,2,BIN 00011111,BIN
00111111,BIN 01001111,BIN 00011111
9220 DATA "m",12,BIN 11001001,BIN 11000000,BIN 10011010,BIN 10111110,BIN
 11111000,BIN 11111100,BIN 11111000
9230 DATA "n",BIN 00111111,BIN 00011111,BIN 00001011,BIN 00001011,BIN 01
000010,BIN 01100011,BIN 01000110,128
9240 DATA "o",BIN 11111000,254,BIN 10110010,BIN 10010000,1,1,3,1
9300 BORDER 1: PAPER 1: INK 7: CLS
9310 PRINT AT 0,11; INK 6; BRIGHT 1; FLASH 1;"DEFENDER"
9315 PRINT
9320 LET c$="Du bist mit deinem Raumjaeger      im Weltraum Stationiert
 um  die Invasion der Gologs abzuweh-ren"
9325 GO SUB 9600
9330 LET c$="Sie greifen mit 3 Verschiedenen Raumschifftypen an"
9335 GO SUB 9600
9340 LET c$="Den Jaegern   "
9345 GO SUB 9600
9350 LET c$="Den Raketen   "
9355 GO SUB 9600
9360 LET c$="Und den Landern   "
9365 GO SUB 9600
9370 LET c$="Erreicht ein Lander den Linken  Bildschirmrand so hast du d
as   Spiel verlohren"
9375 GO SUB 9600
9380 LET c$="Das Spiel ist auch Beendet wenn alle drei deiner Jaeger Abg
esch-ossen oder Gerammt Wurden"
9385 GO SUB 9600
9390 LET c$="Fuer jede Rakete oder jeden      Jaeger der den Bildschirmra
nd   erreicht gibt es Punktabzug"
9395 GO SUB 9600
9400 PRINT '"WEITER MIT IRGENTEINER TASTE"
9410 IF INKEY$="" THEN  GO TO 9410
9415 BEEP .5,10
9420 CLS : PRINT "STEUERTASTEN:": PRINT "a=Hoch"
9430 PRINT "z=Runter"
9440 PRINT "m=FEUER"
9450 PRINT ''Zum Starten Taste Druecken"
9460 IF INKEY$="" THEN  GO TO 9460
9470 CLS : PRINT AT 11,11; FLASH 1;"VIEL GLUECK"
9480 FOR i=1 TO 25: BEEP .01,i: NEXT i: CLS : GO TO 5
9600 PRINT : FOR i=1 TO LEN c$
9610 PRINT c$(i);: BEEP .008,-30+i: NEXT i
9620 RETURN
```

ZX Spectrum

Lui der Wurm



Lui der Wurm

Sie sind Gärtner und müssen „Lui den Wunderwurm", der Sie schon seit geraumer Zeit ärgert, vernichten. Lui aber ist gerissen. Er versteckt sich hinter einer Mauer, wo Sie ihn nicht sehen können. Es bleibt Ihnen also garnichts weiter übrig, als die Mauer mittels einer Laserpistole einzuschießen. Erst jetzt können Sie Lui sehen und natürlich will er sofort ausreißen. Sie dürfen jetzt aber auf keinen Fall einfach schießen. Denn – treffen Sie Lui nicht am Wurmende, sondern in der Mitte oder auch am vorderen Teil, dann wird er bei jedem 3. Fehlschuß länger! Also aufgepaßt! Sie müssen ihre Schüsse gezielt auf Luis Wurmende setzen, um ihn zu vernichten!

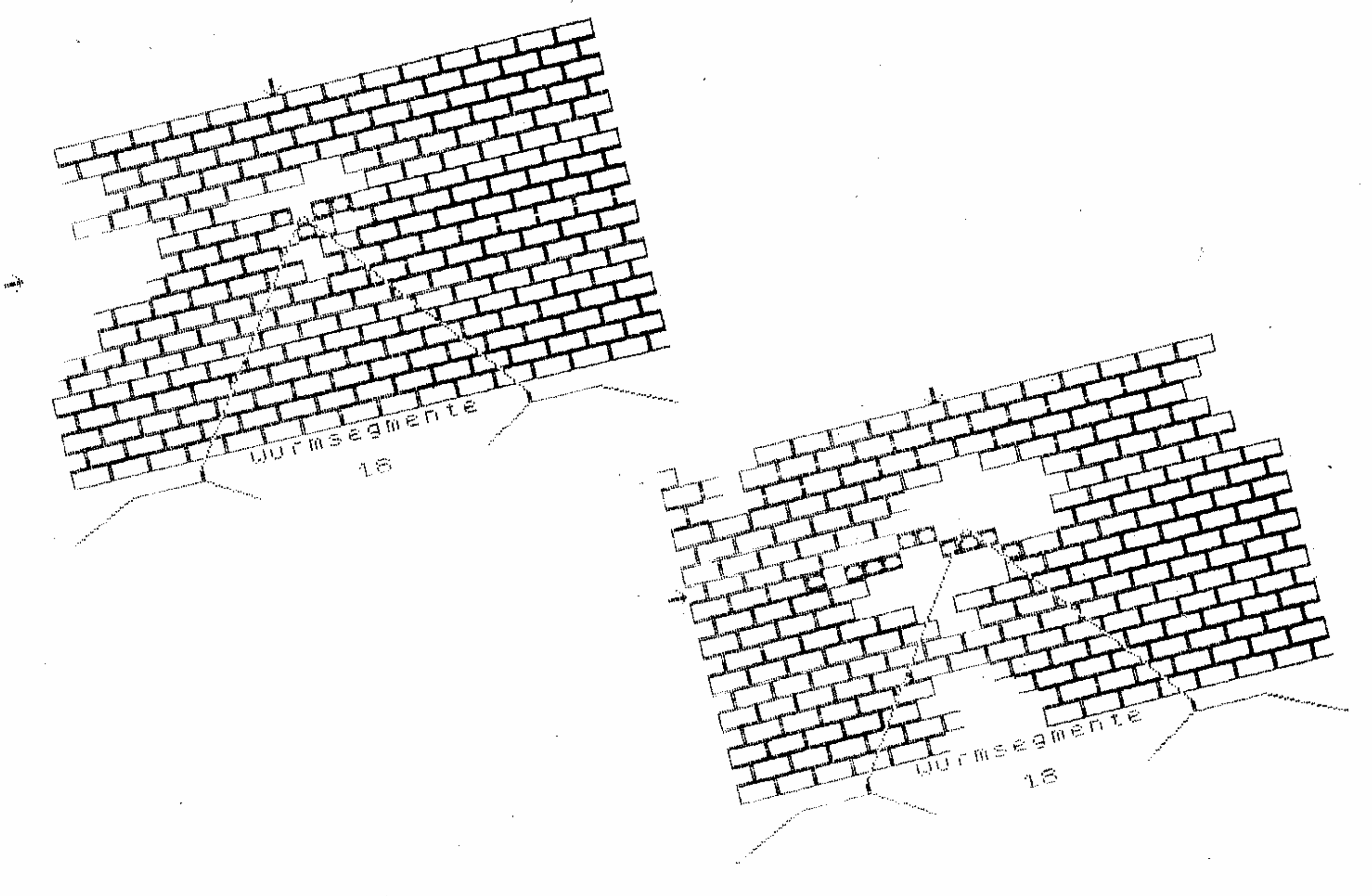
```
   1 REM 0000000000000000000
   2 BRIGHT 1: CLS
   3 REM x by Jens Graefe                          Hochkamp 48
     2000 Oststeinbek
   4 PRINT "Brauchen Sie eine Spielanleitung"
   5 INPUT "Spielanleitung?                (j/n)": PAUSE 0: IF INKEY$="J" TH
EN  GO SUB 8600: CLS
   7 INPUT "Stufe von 1(schwer) bis 6" ;l
   8 IF l<1 OR l>6 THEN  GO TO 5
   9 GO SUB 9000
  10 IF PEEK 23560=105 THEN  LET g=g-1: GO SUB 8000
  15 IF PEEK 23560=112 THEN  LET g=g+1: GO SUB 8000
  20 IF PEEK 23560=113 THEN  LET f=f-1: GO SUB 8000
  23 IF PEEK 23560=122 THEN  LET f=f+1: GO SUB 8000
  25 IF PEEK 23560=109 THEN  GO SUB 8100
  27 POKE 23560,0
  29 BEEP .01,20
  30 LET c=c+(INT (RND*3))-1
  32 LET b=b+1*(b=0)-1*(b=32)
  35 LET c=c+1*(c=1)-1*(c=18)
  40 LET d(a+32*(ek=-1))=c
  50 PRINT AT d(b+32*(es=-1)),b;" " AND NOT ATTR (d(b+32*(es=-1)),b)=104

  55 IF CODE SCREEN$ (c,a)=32 THEN  PRINT AT c,a;"▆"
  70 LET a=a+ek: LET b=b+es
  75 LET ek=ek+2*(a=1)-2*(a=31): LET es=es+2*(b=1)-2*(b=31)
  80 GO TO 10
7999 REM **Anfang der Unterprogramme****
8000 LET g=g+1*(g=0)-1*(g=32)
8005 LET f=f+1*(f=0)-1*(f=19)
8007 BEEP .1,0
8010 PRINT  PAPER 3;AT f1,0;" ";AT 0,g1;" ";AT f,0;"→";AT 0,g;"↓"
8020 LET f1=f: LET g1=g
8099 RETURN
8100  LET h=(g*8)-56: LET i=((21-f)*8)-16
8105 LET h1=193-(g*8)
8110 PLOT 59,19: DRAW  OVER 1;h,i: PLOT 196,19: DRAW  OVER 1;-h1,i
8115: FOR x=1 TO 10: BEEP .002,40: NEXT x
8120 PLOT 59,19: DRAW  OVER 1;h,i: PLOT 196,19: DRAW  OVER 1;-h1,i
8130 IF ATTR (f,g)=104 THEN  PRINT AT f,g;" ": GO TO 8150
8140 IF ATTR (f,g)=120 AND CODE SCREEN$ (f,g)=0 THEN  GO SUB 8200: RETUR
N
8150 LET j=j+1
```

```
8160 IF j=3 THEN  LET b=b-es: LET k=k+1: LET j=0
8170 IF k>18 THEN  LET b=b+es: LET k=k-1
8180 PRINT  INK 9; PAPER 0;AT 21,15;k; PAPER 7;" "
8199 RETURN
8200 IF d(b+32*(es=-1))<>f OR b<>9 THEN  RETURN
8210 FOR x=20 TO -20 STEP -1: BORDER 0: BEEP .01,x: BORDER 7: NEXT x
8215 PRINT AT d(b+32*(es=-1)),9;" "
8220 LET b=b+es: LET k=k-1
8230 PRINT  INK 9; PAPER 0;AT 21,15;k; PAPER 7;" "
8240 IF k=1 THEN  GO TO 8300
8299 RETURN
8300 LET n=FN t()
8305 CLS : GO SUB 8400
8320 DATA .1,3,.1,3,.1,3,.3,0
8321 DATA .1,1,.1,1,.1,1,.3,-2
8322 DATA .1,2,.1,4,.1,6,.4,-10
8330 PRINT AT 5,5; FLASH 1;"Herzlichen glueckwunsch": GO SUB 8400:
8340 PRINT AT 10,0;"Du hast in ";n;" Sec. gewonnen"
8350 GO SUB 8400
8360 PRINT AT 12,0;"Bestzeit ";m$;" ";m;" Sec. ";PEEK 23767;".Stufe"
8365 IF n<m AND PEEK 23767>=1 THEN  GO SUB 8500
8370 PRINT AT 19,0;"Bei Tastendruck gehts weiter": PAUSE 0: RUN
8400 RESTORE 8310
8410 FOR x=1 TO 12: READ y,z: BEEP y,z: NEXT x
8420 RETURN
8500 PRINT "Du hast die neue Bestzeit": BEEP 1,1
8505 INPUT "Name "; LINE m$
8506 LET m$=m$+"      ": LET m$=m$(1 TO 5)
8507 FOR x=23762 TO 23766: POKE x,CODE m$(x-23761): NEXT x
8510 LET p1=INT (n/256)
8520 LET p2=n-(p1*256)
8530 POKE 23760,p1: POKE 23761,p2
8535 POKE 23767,1
8540 PRINT AT 17,0;"Willst Du sie absavèn?"
8550 INPUT "ja/nein";a$
8560 IF a$="ja" THEN  SAVE "Wuermli" LINE 1
8570 RETURN
8600 REM **Spielanleitung**
8610 PRINT "Folgendes Problem:"
8620 PRINT "Sie sind Gaertner und muessen   Lui,den Wunderwurm vernichte
n."
8630 PRINT "Er versteckt sich aber hinter   einer Mauer.Sie muessen jetz
t   zuerst die Mauer einschiessen,  damit Sie Lui erst einmal zu    sehe
n bekommen"
8640 PRINT "Aber Sie duerfen auf keinen Fallnur wild draufloschiesèn,de
nn  a)Wird Lui alle 3 Schuesse um   ein Glied laenger."
8650 PRINT "b)Koennen sie Lui nur kuerzen,  wenn Sie sein LETZTES Glied
    treffen."
8660 PRINT "Es gilt also so zu Schiessen,   dass Sie erkennen koennen,wa
nn  das sichtbare Glied das Letzte  ist.Und das alles geht noch auf Zeit
."
8670 PAUSE 0
8680 CLS : PRINT "Tasten:"'"i/p=links/rechts"'"q/z=rauf/runter"'"m=Schus
s "
8685 PRINT "Also,nun wirds ernst!"'"Zuerst einmal so viel Mauer-    Stue
cke einschiessen,dass Sie   Lui sinnvoll dezimieren koennen."
8690 PAUSE 0: RETURN
8998 STOP
8999  REM **Anfang der Anfangsrutienen**
9000 RESTORE 9000: FOR y=144 TO 148: READ p$: FOR x=0 TO 7: READ z: POKE
```

```
 USR CHR$ y+x,z: NEXT x: NEXT y
9010 DATA"□",255,255,BIN 11000011,129,129,BIN 11000011,255,255
9020 DATA"⊏",255,128,128,128,128,128,128,255
9021 DATA"⊐",255,1,1,1,1,1,1,255
9022 DATA"→",8,4,2,255,255,2,4,8
9025 DATA"↓",24,24,24,24,BIN 10011001,BIN 01011010,60,24
9027 REM ***Einlesen der Anfangswerte fuer die Variablen***
9030 READ a,b,c,ek,es,f,fl,g,gl,j,k
9035 DATA 5,15,10,-1,-1,10,10,15,15,0,10
9040 PLOT 0,0: FOR x=1 TO 6: READ y,z: DRAW y,z: NEXT x
9044 RESTORE 9045: PLOT 255,0: FOR x=1 TO 6: READ y,z: DRAW -y,z: NEXT x

9046 DATA 31,15,28,0,0,4,1,0,0,-4,23,-15
9050 FOR y=1 TO 17 STEP 2: FOR x=0 TO 30 STEP 2: PRINT  PAPER 5;AT y,x;"
⊏⊐";AT y+1,x;"⊐⊏": NEXT x: NEXT y
9052 FOR x=1 TO 18: PRINT  PAPER 3;AT x,0;" ": NEXT x
9054 FOR x=0 TO 31: PRINT  PAPER 3;AT 0,x;" ": NEXT x
9055 DIM d(96)
9057 PRINT  PAPER 3;AT f,0;"→";AT 0,g;"↓"
9060 FOR x=1 TO 96: LET d(x)=16: NEXT x
9070 PRINT AT 19,10;"Wurmsegmente"
9075 PRINT  PAPER 0; INK 7;AT 21,15;k
9100 LET m=256*PEEK 23760+PEEK 23761
9103 FOR x=23672 TO 23674: POKE x,0: NEXT x
9105 LET m$="     "
9110 FOR x=23762 TO 23766: LET m$(x-23761)=CHR$ PEEK x: NEXT x
9120 DEF FN t()=INT ((65536*PEEK 23674+256*PEEK 23673+PEEK 23672)/50)
9190 BEEP 1,10
9800 RETURN
9999 PRINT (PEEK 23641+256*PEEK 23642-PEEK 23635-256*PEEK 23636)-229
```



Altern. Zeichensatz

Alternativer Zeichensatz

Haben Sie schon einmal mit dem Gedanken gespielt, auf Ihrem ZX Spectrum mit einem anderen Zeichensatz zu arbeiten? Wir finden beispielsweise die Kleinbuchstaben von Sinclair nicht besonders schön und waren begeistert, als wir im Handbuch lasen, wie einfach es ist, einen alternativen Zeichensatz einzuschalten: in die Systemvariable CHARS (Speicherstellen 23606 und 23607) braucht nur die Adresse des neuen Zeichensatzes gepoked werden. Doch zuvor müssen in mühsamer Tipparbeit die neuen Zeichen definiert werden.

Der ZX Spectrum-Zeichensatz, der im ROM gespeichert ist, umfaßt alle Zeichen mit den ASCII-Codes 32 (Space) bis 127 (Copyright-Symbol) und ist 1024 Bytes lang. Jedes Zeichen ist in einer Matrix von 8 x 8 Bits, d. h. in 8 Bytes, definiert. Vor dem ersten Zeichen (Space) befinden sich 256 Bytes mit dem Inhalt 255. Wollen wir einen eigenen Zeichensatz benutzen, müssen wir ihn im RAM ablegen und darauf achten, 'daß er den gleichen Aufbau hat wie der Original-Zeichensatz.

Das abgedruckte Programm veranschaulicht, wie in DATA-Statements ein Zeichensatz definiert wird, der zur Darstellung von Schreibschrift benutzt werden kann. An Stelle der Dezimalwerte können Sie auch, wie im Handbuch zur Definition der „User Defined Graphics" dargestellt, Dualzahlen mit der BIN-Funktion eingeben und dadurch das Bitmuster der Zeichen verdeutlichen. Die neuen Zeichen werden oberhalb von RAMTOP gespeichert, so daß sie vor dem Überschreiben durch Basic-Program geschützt sind. Das Programm läuft auf der 16K- und der 48K-Version des Spectrums und legt den Zeichensatz so ab, daß auch noch die „User Defined Graphics" benutzt werden können.

Die Möglichkeit, mit dem Spectrum Schreibschrift darzustellen, ist sicher nur eine reizvolle Spielerei. Doch Sie können ja jederzeit zwischen dem eigenen und dem originalen Zeichensatz durch entsprechende POKE-Befehle umschalten.



* *

```
0>REM Programm zum Erstellen
   eines alternativen
 Zeichensatzes              © 1983 A.&M.Scholz
  10 REM RAM Pruefen (16/48K)
  20 IF PEEK 23732+256*PEEK 23733=65535 THEN  CLEAR 63999: LET char=6400
0: GO TO 40
  30 CLEAR 31487: LET char=31488
  40 REM Zeichensatz laden
  45 CLS : PRINT AT 10,0;"Neuer Zeichensatz wird geladen. ",,"Bitte wart
en.",,,,,,: POKE 23607,250
  50 FOR x=char TO char+255: POKE x,0: NEXT x
  60 FOR x=char+256 TO char+1023 STEP 8
  65 READ a$: PRINT a$;
  70 FOR y=x TO x+7: READ a: POKE y,a: NEXT y
  80 NEXT x
  90 BEEP 1,1: CLS : POKE 23607,60
 100 REM Bedienungsanleitung
 105 POKE 23607,INT (char/256): PRINT TAB 7;"Bedienungsanleitung"; OVER
1;AT 0,7;"___________________"
 110 PRINT AT 2,0;"Geben Sie bitte ein:": POKE 23607,60: PRINT "POKE 236
07,";INT (char/256): POKE 23607,INT (char/256): PRINT AT 5,0;"Das liefer
t Ihnen den neuen     Zeichensatz:";AT 7,0;: GO SUB 200
 120 PRINT AT 11,0;"Mit dem Befehl ";: POKE 23607,60: PRINT " POKE 23607
,60": POKE 23607,INT (char/256): PRINT "wird der normale Zeichensatz
wieder eingeschaltet:": POKE 23607,60: GO SUB 200: POKE 23607,INT (char/
256)
 130 PRINT AT 18,0;"Um den Zeichensatz zu sichern,  wurde ";: POKE 23607
,60: PRINT "RAMTOP ";: POKE 23607,INT (char/256): PRINT "auf ";: POKE 23
607,60: PRINT char-1;: POKE 23607,INT (char/256): PRINT " gesetzt."
 140 PRINT #0;AT 0,0;"Der Zeichensatz kann mit": POKE 23607,60: PRINT #0
;TAB 5;"LOAD name CODE ";char;",1024": POKE 23607,INT (char/256): PRINT
#0;"auf Band gespeichert werden."
 150 POKE 23607,60: PAUSE 0: STOP
 200 REM Anzeige Zeichensatz
 210 FOR x=32 TO 127: PRINT CHR$ x;: NEXT x
 220 RETURN
1000 REM Zeichensatz
1010 DATA " ",0,0,0,0,0,0,0,0
1020 DATA "!",16,16,16,16,0,16,0,0
1030 DATA """",36,36,0,0,0,0,0,0
1040 DATA "#",36,126,36,36,126,36,0,0
1050 DATA "$",8,30,40,28,10,60,8,0
1060 DATA "%",98,100,8,16,38,70,0,0
1070 DATA "&",16,40,16,42,68,58,0,0
1080 DATA "'",4,8,16,0,0,0,0,0
1090 DATA "(",4,8,8,8,8,4,0,0
1100 DATA ")",32,16,16,16,16,32,0,0
1110 DATA "*",0,20,8,62,8,20,0,0
1120 DATA "+",0,8,8,62,8,8,0,0
1130 DATA ",",0,0,0,0,8,8,16,0
1140 DATA "-",0,0,0,62,0,0,0,0
1150 DATA ".",0,0,0,0,24,24,0,0
1160 DATA "/",2,4,8,16,32,64,0,0
1170 DATA "0",24,36,36,36,36,24,0,0
1180 DATA "1",24,40,72,8,8,8,0,0
1190 DATA "2",24,36,4,8,16,62,0,0
1200 DATA "3",24,36,8,4,36,24,0,0
1210 DATA "4",24,40,72,126,8,8,0,0
1220 DATA "5",60,32,56,4,4,56,0,0
1230 DATA "6",12,16,32,44,50,28,0,0
```

"Schreibschrift"

```
Bedienungsanleitung
Geben Sie bitte ein:
POKE 23607,250
Das liefert Ihnen den neuen
Zeichensatz:
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑_
£abcdefghijklmnopqrstuvwxyz{|}~©
Mit dem Befehl  POKE 23607,60
wird der normale Zeichensatz
wieder eingeschaltet:
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑_
£abcdefghijklmnopqrstuvwxyz{|}~©
Um den Zeichensatz zu sichern,
wurde RAMTOP auf 63999 gesetzt.
```

```
1240 DATA "7",60,4,8,28,16,32,0,0
1250 DATA "8",24,36,24,36,36,24,0,0
1260 DATA "9",24,36,36,28,68,56,0,0
1270 DATA ":",0,0,16,0,0,16,0,0
1280 DATA ";",0,16,0,0,16,16,32,0
1290 DATA "<",0,4,8,16,8,4,0,0
1300 DATA "=",0,0,62,0,62,0,0,0
1310 DATA ">",0,16,8,4,8,16,0,0
1320 DATA "?",24,36,4,8,0,8,0,0
1330 DATA "@",0,60,74,86,94,64,60,0
1340 DATA "A",8,20,36,60,164,67,0,0
1350 DATA "B",120,164,56,36,162,124,0,0
1360 DATA "C",152,164,168,112,34,28,0,0
1370 DATA "D",120,164,34,34,226,92,0,0
1380 DATA "E",124,162,56,32,34,28,0,0
1390 DATA "F",124,162,56,32,160,64,0,0
1400 DATA "G",28,34,32,44,34,30,2,28
1410 DATA "H",102,165,60,36,36,102,0,0
1420 DATA "I",60,68,4,4,72,48,0,0
1430 DATA "J",60,68,4,4,52,68,72,48
1440 DATA "K",102,168,48,40,37,194,0,0
1450 DATA "L",152,164,120,32,226,92,0,0
1460 DATA "M",34,54,42,42,162,65,0,0
1470 DATA "N",33,50,42,42,164,68,0,0
1480 DATA "O",25,46,36,36,36,24,0,0
1490 DATA "P",60,82,28,16,80,32,0,0
1500 DATA "Q",24,36,36,36,44,27,0,0
1510 DATA "R",120,164,56,40,164,66,0,0
1520 DATA "S",12,82,60,16,144,96,0,0
1530 DATA "T",122,148,16,16,144,96,0,0
1540 DATA "U",68,164,36,36,36,26,0,0
1550 DATA "V",65,162,36,40,48,32,0,0
1560 DATA "W",65,162,34,42,54,34,0,0
1570 DATA "X",68,170,16,16,170,68,0,0
1580 DATA "Y",34,82,18,18,18,12,34,28
1590 DATA "Z",124,136,16,32,66,252,0,0
1600 DATA "[",14,8,8,8,8,14,0,0
1610 DATA "\",0,64,32,16,8,4,0,0
1620 DATA "]",112,16,16,16,16,112,0,0
1630 DATA "^",16,56,84,16,16,16,0,0
1640 DATA "_",0,0,0,0,0,0,255,0
1650 DATA "£",14,16,124,16,113,46,0,0
1660 DATA "a",0,0,28,36,100,155,0,0
1670 DATA "b",32,32,56,36,36,219,0,0
1680 DATA "c",0,0,28,32,96,157,0,0
1690 DATA "d",4,4,28,36,100,155,0,0
1700 DATA "e",0,24,36,56,32,221,0,0
1710 DATA "f",12,18,18,28,48,223,16,16
1720 DATA "g",0,0,24,36,36,223,4,56
```



```
1730 DATA "h",32,32,56,36,100,163,0,0
1740 DATA "i",8,0,8,24,40,199,0,0
1750 DATA "j",8,0,8,24,40,207,8,48
1760 DATA "k",32,40,48,40,36,227,0,0
1770 DATA "l",12,20,24,16,48,205,0,0
1780 DATA "m",0,0,120,84,84,211,0,0
1790 DATA "n",0,0,56,36,36,227,0,0
1800 DATA "o",0,0,28,34,34,220,0,0
1810 DATA "p",0,0,56,36,100,163,32,32
1820 DATA "q",0,0,28,36,100,157,6,4
1830 DATA "r",0,0,20,56,80,159,0,0
1840 DATA "s",0,0,16,56,68,155,0,0
1850 DATA "t",8,30,8,24,40,199,0,0
1860 DATA "u",0,0,36,36,36,219,0,0
1870 DATA "v",0,0,34,36,40,215,0,0
1880 DATA "w",0,0,68,84,84,171,0,0
1890 DATA "x",0,0,54,8,20,227,0,0
1900 DATA "y",0,0,36,36,100,159,4,24
1910 DATA "z",0,0,56,72,144,11,4,56
1920 DATA "{",14,8,48,8,8,14,0,0
1930 DATA "|",8,8,8,8,8,8,0,0
1940 DATA "}",112,16,12,16,16,112,0,0
1950 DATA "~",20,40,0,0,0,0,0,0
1960 DATA "©",60,66,153,161,161,153,66,60
```

ZX Spectrum

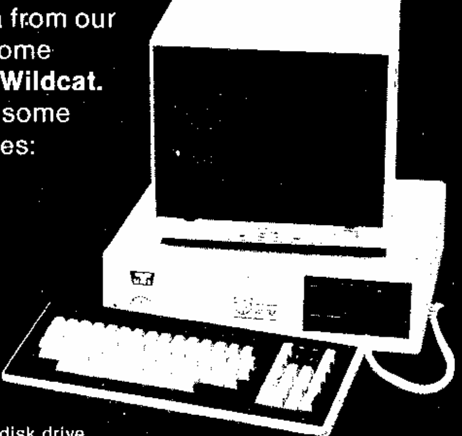
WILDCAT™

Looks Like an IBM™ Works Like an Apple™

To a land where fruit and flowers reign supreme, comes the awesome power of the **Wildcat** to challenge their supremacy.

Wildcat is a sleek styled mobile computer designed for your vehicle, boat or aircraft. It looks like an IBM PC™ and is fully software compatible with the Apple II™ product line at a list price that would put a smile on the face of the most frugal computer buyer.

But price isn't everything. All those features that would cost you hundreds of dollars extra from our competitors, come standard with **Wildcat.** Let's compare some of these features:



	Apple IIe	Wildcat
Detachable keyboard	No	STD
Full numeric key pad	Option	STD
Full functions keys	No	41
Built in disk controller	No	STD
Parallel printer port	No	STD
RS 232 serial port	No	STD
Game port	1	2
RGB video out	Option	STD
Composite video	STD	STD
RF video for TV	Option	STD
CP/M	Option	STD
Hi Res graphics (6 color)	STD	STD
Low Res graphics (16 color)	STD	STD
64KB memory	STD	STD
Half high disk drives	No	STD
Converters for vehicles, boats, and aircraft	No	Option
Aluminum carrying case	No	Option
List price	$1940*	$1099.00*

For more information on the all new **Wildcat**, see your local computer dealer or call or write:



COMPUTER AND PERIPHERAL PRODUCTS
1530 S. Sinclair
Anaheim, CA 92806
(714) 978-9820

*Computer plus one disk drive

Warning: This equipment is exempt from compliance with FCC testing requirements pursuant to 47 CFR 15.801 (c) (1)
Operation of this equipment in a residential area may cause interference.

CIRCLE 181 ON READER SERVICE CARD

VC-64

Autostart

Autostart

benötigte Hardware: CBM-64 & Floppy VC 1541

Mit diesem Programm können Programme auf Diskette in Autostart-Programme umgewandelt werden, die nach dem Laden automatisch starten.

Das Programm erzeugt auf der Diskette einen Programmfilm mit der Bezeichnung „AUTO NAME", welches mit LOAD „AUTO NAME", 8,1 zu laden ist und dann automatisch startet. Die Anfügung eines „,1" hinter dem LOAD-Befehl ist wichtig, da das Programm absolut geladen werden muß. Autostart läßt sich übrigens gut zum Listschutz verwenden, indem man dem Anfang des Programms O POKE 788,52:770,148:771,227 hinzufügt. Diese Zeile verriegelt die RUN/STOP und RESTORE-Taste und veranlaßt den C-64 dazu, in den START-Zustand zurückzuspringen.

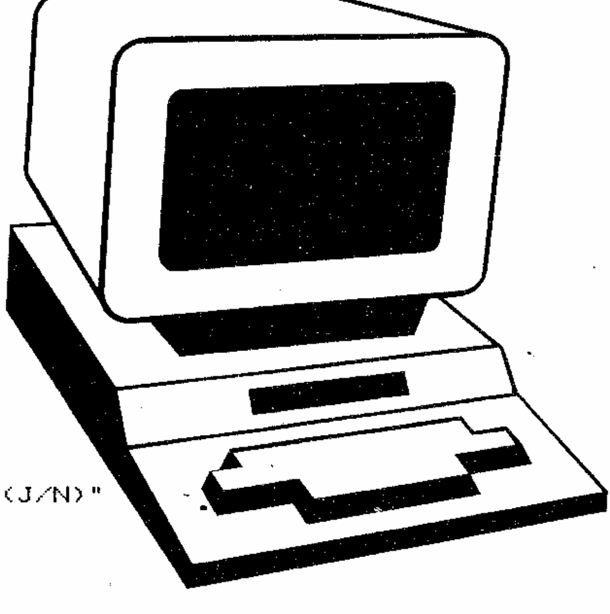
Durch den Zusatz ',1' beim LOAD-Befehl wird das Programm „AUTO NAME" ab der Adresse 804 geladen. Dadurch wird die Adresse der INPUT-Routine des Interpreters (804,805) verändert. Sie zeigt nun auf ein kleines Maschinenprogramm im Tastaturpuffer.

Nach dem Beenden des Ladevorgangs springt der Interpreter nun zu dem Maschinenprogramm im Tastaturpuffer, anstatt zur normalen Interpreterroutine.

Das MC-Programm bewirkt nun die Wiederherstellung des normalen INPUT-Vektors. Anschließend wird der Befehl RUN simuliert und dadurch das Programm gestartet.

Zeilen	Funktion
0– 100	kurze Erklärung und Eingabe des Filenamens
100– 500	Überprüfung, ob das File den Anforderungen entspricht
500– 700	Erstellen des neuen Programmfiles
5000– 6000	Abfrage des Disketten-Fehlerkanals
10000– 11000	MC-Routine (siehe oben)

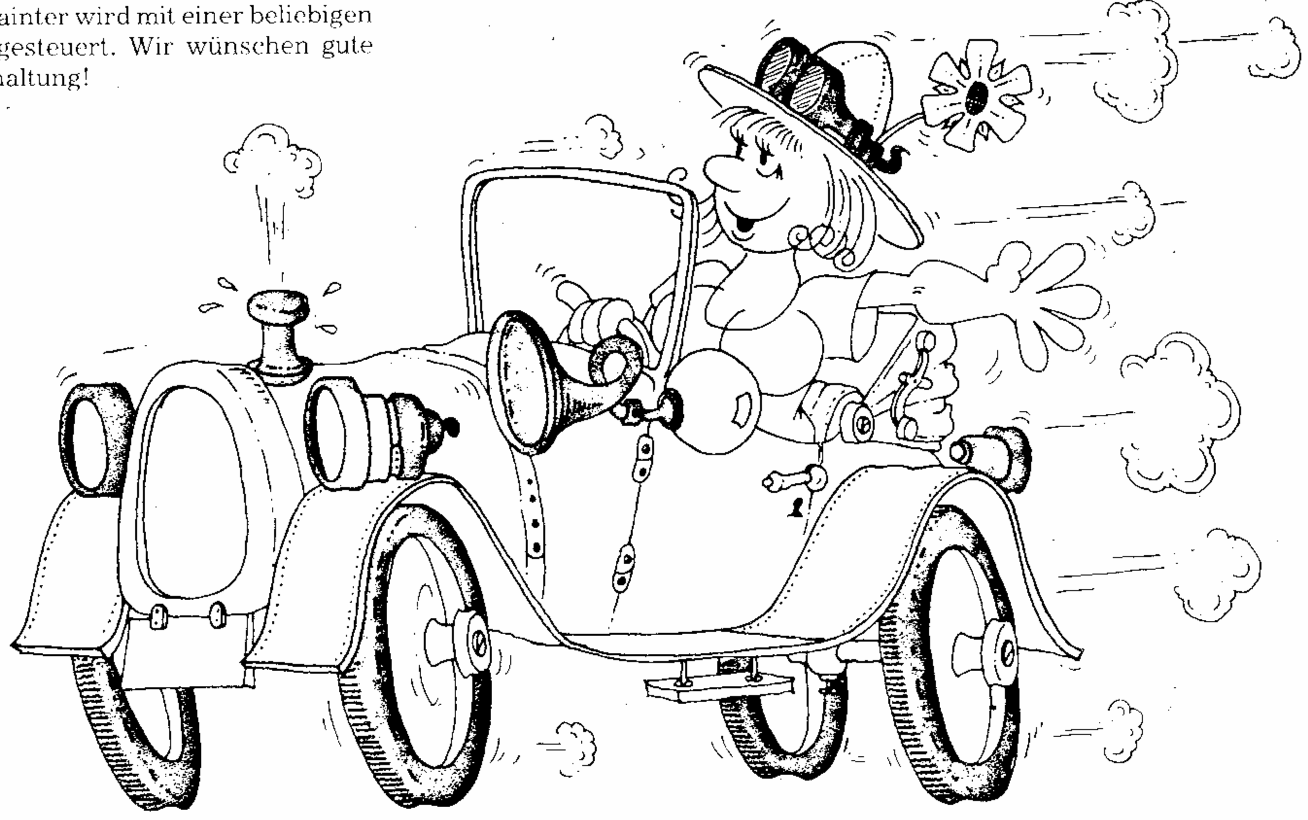
```
10 REM      *** AUTO START ***
20 POKE53280,5:POKE53281,5
30 PRINT"*** AUTO START ***"
40 PRINT"(C) 1984 BY LIPPE ENTERPRISES"
50 PRINT"DIESES PROGRAMM WANDELT EIN PROGRAMM AUF";
60 PRINT"DER DISKETTE IN EIN AUTOSTART PROGRAMM"
70 PRINT"UM. ES IST NACHHER MIT LOAD"CHR$(34)"AUTO NAME"CHR$(34)",8,1";
75 PRINT" ZU LADEN."
80 PRINT"BITTE GEBEN SIE JETZT DEN FILENAMEN EIN:"
100 INPUTN$
105 IFLEN(N$)>16THENPRINT"NAME ZU LANG (MAX. 16 ZEICHEN)":GOTO170
110 OPEN1,8,0,"0:"+N$+",P,R"
120 GOSUB5000
130 OPEN1,8,0,"0:"+N$+",P,R":GET#1,A$,B$:IFA$=CHR$(1)ANDB$=CHR$(8)THEN500
140 PRINT"DAS ZU AENDERNDE PROGRAMM IST KEIN"
150 PRINT"PROGRAMM, DAS VOM BASIC-ANFANG AN"
160 PRINT"GELADEN WIRD."
170 PRINTTAB(30)"RETURN"
180 POKE198,0:WAIT198,1:POKE198,0:RUN
500 OPEN2,8,1,"0:AUTO "+N$+",P,W"
520 PRINT#2,CHR$(36)CHR$(3);
525 PRINT#2,CHR$(128)CHR$(3);
530 FORX=806TO827:PRINT#2,CHR$(PEEK(X));:NEXT
540 FORX=828TO895:PRINT#2,CHR$(32);:NEXT
550 FORX=896TO930:READA:PRINT#2,CHR$(A);:NEXT
560 FORX=931TO2047:PRINT#2,CHR$(32);:NEXT
570 PRINT#2,CHR$(0);
580 GET#1,A$:A$=LEFT$(A$+CHR$(0),1):IFST<>0THEN700
590 PRINT#2,A$;:GOTO580
700 PRINT#2,A$:CLOSE2
710 GOSUB5000
720 PRINT"NOCH EINMAL? (J/N)"
730 GETA$:IFA$="J"THENRUN
740 IFA$="N"THENSYS64738
750 GOTO730
5000 OPEN3,8,15:INPUT#3,C$,D$,E$,F$:CLOSE3:CLOSE1
5005 IFC$="00"ORC$="01"THENRETURN
5010 PRINT"DISK ERROR:"C$" "D$" "E$" "F$
5040 OPEN3,8,15,"S0:AUTO "+N$:CLOSE3:GOTO170
10000 REM MC CODE
10001 DATA169,87,141,36,3,169,241,141,37,3,169,4,133,198,169,82,141,119,2,169
10002 DATA85,141,120,2,169,78,141,121,2,169,13,141,122,2,96
READY.
```



Roadpainter

Roadpainter
Ein kleines blaues Auto „Der Roadpainter" treibt in einem kleinen Dorf sein Unwesen. Er bemalt die Straßenstücke, die er überfahren hat mit grünen Kreisen und malt diese bei einer weiteren Berührung aus. Das wäre natürlich sehr einfach, wenn da nicht dieses Auto wäre, das in entgegengesetzter Richtung fährt. Sie müssen nun versuchen, mit viel Reaktion und guter Taktik nicht mit diesem Auto zu kollidieren. Wenn sie es geschafft haben sollten, die gesamte Straße zu bemalen, wird der restliche Bonus zu ihrem Score addiert.
In der 1. Runde sind die Kreise bereits vorgezeichnet. Es genügt also ein einmaliger Durchlauf. Mit jeder weiteren Runde erhöht sich die Anzahl der Felder, die noch nicht bemalt sind. Ab der vierten Runde erscheint eine vollkommen neue Landschaft mit einer noch komplizierteren Strecke. Der Roadpainter wird mit einer beliebigen Taste gesteuert. Wir wünschen gute Unterhaltung!

Einige Details zum Programm
Dieses Programm wurde für den C-84 + Floppy Disc geschrieben.
Um es auch mit einer Datasette spielen zu können ist folgendes zu ändern.
Zeile
41005 Open2,1,1,„CARDATA"
42000 Open2,1,0:„CARDATA"
Weitere Besonderheiten
Bei der ersten Inbetriebnahme des Programms muß in Zeile 150 „Gosub 42000" entfernt werden, da die Datei mit der Bestenliste noch nicht existiert und es somit zu einer Fehlermeldung führen würde. Wurde das Spiel dann einmal gespielt, muß das „Gosub 42000" wieder eingesetzt werden. Beim Eintippen des Programms sollte man den Zeilen 56100–58120 besondere Beachtung schenken, da diese sehr viele (unvermeidbare) Steuerzeichen enthalten. Bei den Datas auf die vielen Beistriche achten.
Vor dem Starten des Programms sollte dieses gesaved werden, da ein Fehler in den Datas der Maschinenprogramme zu einem Absturz des Systems führen kann. Nach dem Starten dauert es etwa eine Minute bis auf dem Monitor etwas Sichtbares passiert.
Falls das Programm vorzeitig unterbrochen werden soll muß man die Tasten Runstop/Restore drücken, da die Stoptaste abgeschaltet ist.



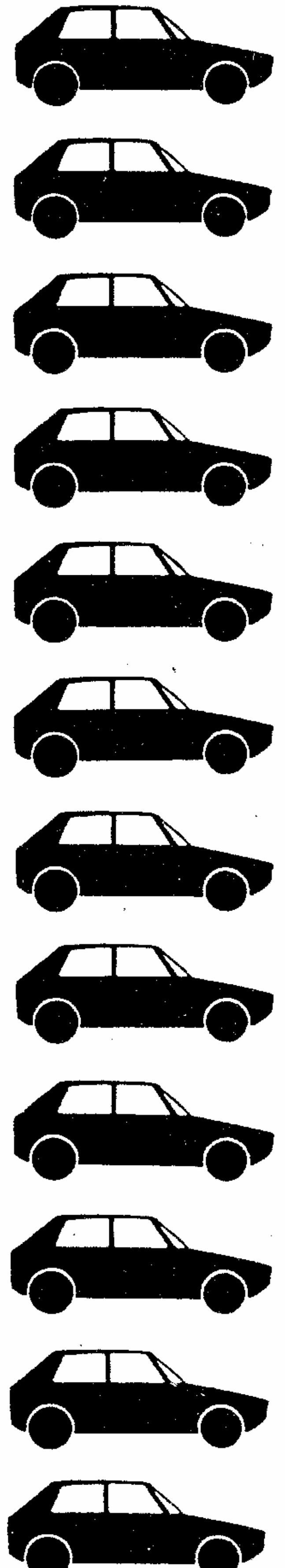
```
1 REM ROADPAINTER
2 REM (C) BY RONALD MAYER
3 REM THENING 24
4 REM 4062 THENING
5 REM
6 REM MADE IN AUSTRIA
7 REM
100 V=53248:BI=V+33:RA=BI-1:S=54272:F=1024:D1=20000:X=1
105 POKE788,52:POKE56,56
110 POKE2040,255:POKEV+29,1
115 POKEV+39,7:POKEV+40,3:POKEV+41,14:POKEV+34,0:POKEV+35,9
120 DIMML(19),MH(19),MF(19),PU(16),NA$(16)
125 FORI=0TO5:READB(I),C(I),D(I),S(I):NEXT
130 FORI=0TO8:READSS(I):NEXT
135 DATA-40,1,-8,15,-1,,-8,13,40,1,8,11,1,,8,14,-40,1,-8,15,-1,,-8,13
140 DATA248,268,288,316,346,376,406,436,466
145 X$="[illegible]"
150 GOSUB60000:GOSUB42000:GOSUB58000
5000 POKEBI,6:POKERA,S
5010 PRINT"[illegible]"TAB(15)"ROADPAINTER":POKE53272,30
5020 PRINT"[illegible]PLAYER"X"[illegible] ← ";NB$(X)"                    "
5025 GETA$:IFA$="+"ANDX<10THENX=X+1:GOTO5020
5027 IFY>1000THENY=0:GOSUB58000:GOTO5000
5030 IFA$="-"ANDX>1THENX=X-1:GOTO5020
5035 IFA$=CHR$(13)ANDNB$(X)<>""ANDNB$(X)<>"?"THENNA$(0)=NB$(X):GOTO10000
5040 IFA$<>CHR$(13)THENY=Y+1:GOTO5025
5045 INPUT"[illegible]ENTER YOUR NAME";NB$(X)
5047 IFNB$(X)=""THENPRINT"[illegible]";:GOTO5045
5050 NA$(0)=NB$(X):GOTO10000
10000 CA=3:M=3:AG=0:PU=0:SR=0
10003 GOSUB55000
10006 POKE2042,13:POKE2041,14
10007 POKEV+4,143:POKEV+5,58
10008 POKEV+2,159:POKEV+3,58
10009 IFM>0THENGOSUB50000
10010 P=1:PO=1079:PQ=1081
10012 POKEV+30,0:POKEF+753,47+CA:Q=3
10014 POKEF+953,49+AG
10025 IFPEEK(PO+B(P))>46THENP=P+1:POKE2042,S(P):IFP>4THENP=1
10027 IFPEEK(V+30)=6THEN25000
10030 PO=PO+B(P)
10033 IFPEEK(PQ+B(Q))>46THENQ=Q-1:POKE2041,S(Q):IFQ<1THENQ=4
10035 SYS32839
10036 IFPEEK(PO)<46THENSYS36826:POKEPO,PEEK(PO)+1:SR=SR+1:PU=PU+1:GOTO10053
10040 SYS36886
10053 IFSR=SS(AG)THEN20000
10055 IF(PEEK(PQ+B(Q-1))=0ORPEEK(PQ+B(Q+1))=0)THENGOSUB15020
10060 PQ=PQ+B(Q):SYS32839:SYS36886
10061 IFPEEK(203)=64THEN10025
10062 IFPEEK(PO+B(P+1))=0THENW=P+1:GOTO15000
10063 IFPEEK(PO+B(P-1))=0THENW=P-1:GOTO15000
10070 GOTO10025
15000 POKEV+4+C(W),PEEK(V+4+C(W))+D(W)*2:PO=PO+B(W)*2:GOTO10025
15020 IFRND(1)>.5THENRETURN
15025 Z=Q-1:IFPEEK(PQ+B(Q+1))=0THENZ=Q+1
15030 POKEV+2+C(Z),PEEK(V+2+C(Z))+D(Z)*2:PQ=PQ+B(Z)*2:RETURN
20000 SR=0:BO=0:FORI=0TO3:BO=BO+(PEEK(1579-I)-48)*10↑I:NEXT:AG=AG+1
20002 IFBO=0THENM=0:PRINTX$"█NO BONUS":GOSUB32000:GOTO20008
20003 POKE36354,31:FORI=1TOBO
20004 BQ=1-BQ:IFBQ=1THENSYS36938
20005 PU=PU+1:SYS36886:SYS36864:NEXT:POKE36854,11
```

```
20008 PRINTX$"■ ROUND ";AG+1
20009 M=1:GOSUB32000
20012 IFAG/3<>INT(AG/3)THEN20015
20013 PRINTX$"■BONUS CAR"
20014 GOSUB32000:CA=CA+1
20015 GOTO10003
23000 FORI=1TO15:IFPU>PU(I)THEN23020
23015 NEXTI
23020 LP=I:FORII=14TOISTEP-1
23025 PU(II+1)=PU(II):NA$(II+1)=NA$(II):NEXT:PU(I)=PU:NA$(I)=NA$(0):RETURN
25000 FORI=0TO24:POKES+I,0:NEXT
25005 POKEV+30,0:CA=CA-1
25006 POKES+1,30:POKES+5,15:POKES+4,129
25007 FORI=15TO0STEP-.2:POKES+24,I:NEXT .
25010 POKES+4,0:FORI=1TO1000:NEXT
25013 IFCA=0THEN30000
25030 M=0:GOTO10006
30000 PRINTX$"■GAME OVER"
30010 M=2:GOSUB32000
30015 GOSUB23000:GOSUB40000:POKE198,0:WAIT198,1:GOTO5000
32000 FORI=148TO210STEP.25:POKEV,I:NEXT
32003 IFM=0THENFORI=1TO2000:NEXT:GOTO32010
32005 GOSUB50000
32010 FORI=210TO148STEP-.25:POKEV,I:NEXT
32015 PRINTX$"■.........":RETURN
40000 PRINT"■":POKEV+21,0:POKEBI,5:POKERA,2:POKE646,6
40010 PRINT"■■■■■■■■■    ROADPAINTER  HIGHSCORES   ■"
40015 PRINT"■■":TB=1:FORI=1TO15
40020 IFI>9THENTB=0
40023 IFLP=ITHENPOKE646,2
40025 PRINTTAB(4+TB)I"■ ← ";NA$(I);
40030 PRINTTAB(33-LEN(STR$(PU(I))))PU(I)"■":NEXT
40040 IFPU(15)>PUTHENRETURN
41005 OPEN2,8,2,"@:CARDATA,S,W"
41010 FORI=1TO10
41013 IFNB$(I)=""THENNB$(I)="?"
41015 PRINT#2,NB$(I):NEXT
41020 FORI=1TO15
41023 IFNA$(I)=""THENNA$(I)="?"
41025 PRINT#2,NA$(I)
41030 PRINT#2,STR$(PU(I))
41035 NEXT:CLOSE2:RETURN
42000 OPEN2,8,2,"CARDATA,S,R"
42010 FORI=1TO10
42015 INPUT#2,NB$(I):NEXT
42020 FORI=1TO15
42023 INPUT#2,NA$(I)
42030 INPUT#2,A$:PU(I)=VAL(A$)
42035 NEXT:CLOSE2:RETURN
50000 FORI=0TO24:POKES+I,0:NEXT
50010 POKES+24,15
50015 POKES+5,2:POKES+6,242
50020 POKES+12,1:POKES+13,240
50025 POKES+19,12:POKES+20,0
50030 FORJ=1TOM:FORI=1TO19
50035 POKES+4,33:POKES+11,33:POKES+18,17
50040 POKES,ML(I):POKES+1,MH(I)
50045 POKES+7,ML(I)-3:POKES+8,MH(I)
50050 POKES+14,0:POKES+15,10
50055 FORII=1TOMP(I)*40:NEXTII:POKES+18,0:POKES+11,0:POKES+4,0:NEXTI,J
50065 FORI=1TO4:POKES+18,129:POKES+14,200:POKES+15,255:FORII=1TO30:NEXTII
50070 POKES+18,0:FORII=1TO15:NEXTII:NEXTI
```

```
50075 RETURN
55000 POKEBI,13:POKERA,8:POKEV+21,1:POKEV,148:POKEV+1,164
55005 IFAG>2THENGOSUB57000:GOTO55100
55010 GOSUB58000
55100 POKEV+21,7
56100 IFPU>0THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"TAB(31)"▪"1000+AG*200:RETURN
56122 PRINT"▒▒";TAB(32)"▒▒HIGH  "
56123 PRINT"▒▪";TAB(32)RIGHT$("000000"+MID$(STR$(PU(1)),2),6)
56124 PRINT"▒▒";TAB(32)"▒▒SCORE "
56126 PRINT"▒";TAB(32)"▪000000"
56127 PRINT"▒▒";TAB(32)"▒▒BONUS "
56128 PRINT"▒";TAB(32)"▪1000"
56129 PRINT"▒▒";TAB(32)"▒▒CARS  "
56130 PRINT"▒";TAB(32)"▪03"
56131 PRINT"▒▒";TAB(32)"▒▒ROUND "
56132 PRINT"▒";TAB(32)"▪01"
56200 RETURN
57000 PRINT"▒▪  ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬  ▬▬▬▬▬  "
57015 PRINT"   ├────────────────────────┤  ├───┤   "
57020 PRINT" ▬├▒─●───@:───────────▒─┤▬├─┬─┤▬ "
57025 PRINT" ├────┤──────────────────────────────┤ "
57030 PRINT" ├▒─▒─├─┌▌//▌┼────●─────▒─▒─▒▒─▒─┤ "
57040 PRINT" ├:─├─├─▌▒"#"▌├───├────────├─├─├─┤"
57045 PRINT" ├@─├─├─▌▒_▌▌├▒─├─▒─▒─▒─├─├─├─┤ "
57050 PRINT" ├├─├─├─▌▒_▌ ▌├├─├──────├─├─├─┤ "
57055 PRINT" ├├─╥─├─└▬▬▬├─╥─/▌/////▌\─├─├─├─┤ "
57060 PRINT" ├├────├──────────▌▒"###"▌├─├─├─┤ "
57065 PRINT" ├●────,────────▒┬─┬"▒_$$_▌├─├─├─┤ "
57070 PRINT" ├├──────────├─┤   ▒&   ▌├─├─├─┤ "
57075 PRINT" ├├─▒/▌──────▒▌─├─▌▌─ ▒&  ▌▌"▌├─├─├─@─┤ "
57080 PRINT" ├├─├─────├─├─┤  ▒_  ▌"▌├─├─├─┤ "
57085 PRINT" ├├─├─▒─▒─╥─╥─┤▬▬▬▬▬▬▬├─╥─╥─├─┤ "
57090 PRINT" ├├─├────────────────────────├─┤ "
57095 PRINT" ├├─/"▌//▌┼─▒─▒─,──────┬─▒─┬"\─┤ "
57100 PRINT" ├├─▌▒"#"▌├─├─├────────├─├─▌▌_▌├─┤ "
57105 PRINT" ├├─▌▒_/▌├─├─├─┌──┐─├─├─┤  ├─┤ "
57110 PRINT" ├├─▌▒ _▌├─├─:─┤ ▌_▌ ├─├─┤  ├─┤ "
57115 PRINT" ├├─└▬▬┘─├─├─@─└▬▬├─╥─├─┤  ├─┤ "
57120 PRINT" ├├───────├─├────────├─┤  ├─┤ "
57125 PRINT" ├─▒▌───:@────▒▌─├─▒▌──────▒─┬"\─└▬┘─┤ "
57130 PRINT" ├──────────├────────┤  ├─────┤ "
57135 PRINT"▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▬▬▬▬▬▬▬▬▬▬▬▬  ▬▬▬▬▬▬▬ ▒"
57140 IFAG=3THENRETURN
57145 FORI=1TO30*(AG-3)
57150 ZU=INT(RND(1)*920)+1065
57155 IFPEEK(ZU)<>45THEN57150
57160 POKEZU,44:NEXT
57200 RETURN
58000 PRINT"▒▪ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ "
58015 PRINT" ├────────────────────────────────┤ "
58020 PRINT" ├─▒▌/▌───@:────●────────@:────────▌/▌┐─┤ "
58025 PRINT" ├├────────├────────────────▌▒"▌├─┤ "
58030 PRINT" ├├─┌▌////▌┐─├─┌──────▌///▌"┐─▌▒_▌├─┤ "
58035 PRINT" ├├─▌▒"##"▌├─├─▌▌▒▒▒▒▌+ ▒"#┐▌├─▌▌ ▌├─┤ "
58040 PRINT" ├├─▌▒_$└/▌├─├─▌▒"#"▌├ ▒_┘▌├─▌▌"▌├─┤ "
58045 PRINT" ├:─▌▌ ▒&▌├─├─▌▒_$┘ ▒&▌ ▌▌├─▌▌"▌├─┤ "
58050 PRINT" ├@─┤  ▒_▌├─├─┤   ▒_┘ ▌├─▌▒▒▌├─┤ "
58055 PRINT" ├├─▌▌   ▌├─├─▌▒▒▌▌    ▌├─▌▒"▌├─┤ "
58060 PRINT" ├├─▌▌ ├─ ▌├─├─▌▒"┐ ▌_   ▌├─▌▒&▌├─┤ "
58065 PRINT" ├├─▌▌ "" ▌├─├─▌▒&▒▒▒▌▌ ├─ ▌├─▌▒&▌├─┤ "
58070 PRINT" ├├─└▬▬▬┘─├─▌▒"#┐▌ "" ▌├─▌▒&▌├─┤ "
```

```
58075 PRINT" ├ |---------|-▌▒L$$┘ ▌├    ▪ ├-▌▒L__▪ ├▌"
58080 PRINT" ├●─────────●─┤▬▬▬▬▬▬▬▬▬▬ ├─┤▄▄▀┤▌"
58085 PRINT" ├'|-------------------------|-▌"
58090 PRINT" ├ |- ┌────▌╱╱▪┴─┬▔▔▔▔▔▔▔▔▔┴─┬┐- |-▌"
58095 PRINT" ├ |-▌▒▒╱╱╱╱▒▒▔▪ ├▌  ▒_  _  _ ▪├▌ ├:-▌"
58097 PRINT" ├ |-▌▒▔###▔╱▪ ├ └▬▬▬▬▬▬▬▬▬▬┘-▌ ├@-▌"
58100 PRINT" ├ |-▌▒L$$$$_▪ ├──────────────▌ ├ |-▌"
58105 PRINT" ├ |- └▬▬▬▬▬▬♠───────:@──────○┘- |-▌"
58107 PRINT" ├ |-------------------------|-▌"
58110 PRINT" ├▒'▪──────:@─────────────────────▒'▒─▌"
58115 PRINT" ├───────────────────────────────▌"
58120 PRINT" ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▒"
58130 IFAG=0THENRETURN
58135 FORI=1TO20*AG
58140 ZU=INT(RND(1)*920)+1065
58145 IFPEEK(ZU)<>45THEN58140
58150 POKEZU,44:NEXT
58200 RETURN
59000 POKEBI,2:POKERA,2:POKE53272,22
59006 FORI=0TO24:POKES+I,0:NEXT
59008 POKES+24,15:POKES+6,240
59010 FORI=0TO123:PRINTCHR$(PEEK(D1+I));
59011 IFPEEK(D1+I)<48THEN59014
59012 POKES+1,100:POKES+4,129
59013 IFPEEK(203)=60THENPOKES+4,0:RETURN
59014 FORII=1TO99:NEXT:POKES+4,0:NEXT
59030 FORI=1TO1000:IFPEEK(203)=60THENRETURN
59040 NEXT:GOTO59010
60000 REM SPRITEDATAS
60005 DATA3,128,,115,128,,109,64,,255,192,,255,192,,109,64,,115,128,,3,128
60010 DATA,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
60015 DATA112,,,115,128,,173,128,,255,192,,255,192,,173,128,,115,128,,112
60020 DATA,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
60025 DATA28,,,127,,,127,,,93,,,62,,,221,128,,255,128,,221,128,,62
60030 DATA,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
60035 DATA62,,,221,128,,255,128,,221,128,,62,,,93,,,127,,,127,,,28
60040 DATA,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
60041 DATA0,0,0,0,0,0,0,0,0,255,255,255,127,255,254
60042 DATA63,255,252,63,255,252,28,60,56,28,60,56,31,255,248,31,255,248
60043 DATA28,60,56,28,60,56,63,255,252,63,255,252,127,255,254
60044 DATA255,255,255,,,,,,,,,,,,
60045 FORI=0TO62:READA:POKE832+I,A:NEXT
60050 FORI=0TO62:READA:POKE896+I,A:NEXT
60055 FORI=0TO62:READA:POKE960+I,A:NEXT
60060 FORI=0TO62:READA:POKE704+I,A:NEXT
60063 FORI=0TO62:READA:POKE16320+I,A:NEXT
60100 DATA29,69,34,207,39,18,43,219,46,118
60110 FORI=1TO5:READA,B:HI(I)=A:LO(I)=B:NEXT
60115 DATA2,2,5,5,4,4,3,1,1,2,3,2,5,5,4,4,3,5,1
60120 FORI=1TO19:READA:MH(I)=HI(A):ML(I)=LO(A):NEXT
60125 DATA2,1,2,2,2,2,1,2,2,.6,.6,.6,2,2,2,2,1,2,2
60130 FORI=1TO19:READMP(I):NEXT
61000 FORI=D1TOD1+123:READA:POKEI,A:NEXT
61010 DATA147,5,13,13,29,29,114,111,97,100,112,97,105,110,116,101,114,13,13
61020 DATA13,29,29,99,79,80,89,82,73,71,72,84,29,49,57,56,52,13,13,13,29,29
61030 DATA30,97,29,103,65,77,69,29,66,89,29,114,79,78,65,76,68,29,109,65
61040 DATA65,65,74,69,82,,,,,,,,,,,,,,,,,,,,,,,,,,20,20,20,20,20,89,69,82
61050 DATA158,13,13,13,13,13,13,13,13,29,29,112,82,69,83,83,29,29,62,115
61060 DATA112,97,99,101,60
62000 POKE56334,0:POKE1,51
62010 FORI=8TO1712:POKE14336+I,PEEK(53248+I):NEXT
```

```
62015 POKE1,55:POKE56334,1
62020 POKE53272,30:POKE53270,216
62025 READB
62027 A=PEEK(63):IFA=200THEN63000
62030 FORI=0TO7:POKE14336+A*8+I,B:READB:NEXT:GOTO62027
62208 DATA255,255,195,219,219,195,255,255
62238 DATA255,255,255,231,231,255,255,255
62243 DATA255,191,187,250,255,191,187,119
62244 DATA221,187,191,255,250,187,191,255
62245 DATA250,189,255,233,238,253,239,237
62246 DATA250,191,255,190,191,191,250,255
62247 DATA190,122,191,127,190,122,191,127
62249 DATA,,48,,,,,12
62250 DATA,,3,,,,,
62251 DATA192,,,,,,48,
62252 DATA255,255,255,255,255,255,255,255
62253 DATA255,255,231,219,219,231,255,255
62254 DATA255,255,231,195,195,231,255,255
62255 DATA85,85,85,0,255,255,255,255
62256 DATA60,102,102,102,102,102,60,0
62266 DATA255,255,195,219,219,195,255,255
62272 DATA255,255,255,0,0,255,255,255
62273 DATA7,3,1,,,255,255,255
62274 DATA3,15,63,254,255,55,61,249
62275 DATA,240,252,255,223,239,124,252
62276 DATA254,251,63,62,63,15,0,0
62277 DATA127,255,239,254,252,240,192,0
62281 DATA255,255,255,224,224,225,227,231
62282 DATA231,194,129,,,,,
62284 DATA233,238,247,190,219,255,123,255
62285 DATA231,199,135,7,7,7,7,7
62286 DATA231,227,225,224,224,224,224,224
62287 DATA191,123,191,222,187,230,250,237
62288 DATA250,189,255,231,190,219,175,123
62289 DATA231,227,225,224,224,225,227,231
62290 DATA,40,190,190,190,190,40,0
62291 DATA255,255,255,0,0,129,195,231
62293 DATA231,195,129,0,0,255,255,255
62295 DATA224,192,128,,,255,255,255
62298 DATA231,199,135,7,7,255,255,255
62299 DATA255,255,255,0,0,1,3,7
62300 DATA255,255,255,7,7,255,255,255
62301 DATA231,231,231,231,231,231,231,231
62302 DATA231,231,231,231,231,255,255,255
62303 DATA7,7,7,7,7,135,199,231
62310 DATA255,255,255,224,224,255,255,255
62312 DATA255,255,255,231,231,231,231,231
62313 DATA255,255,255,0,0,128,192,224
62315 DATA7,7,7,7,7,255,255,255
62316 DATA,,,,,1,3,7
62317 DATA224,224,240,248,252,255,255,255
62318 DATA255,255,255,63,31,15,7,7
62320 DATA255,255,255,252,248,240,224,224
62321 DATA255,255,255,7,7,7,7,7
62322 DATA255,255,255,224,224,224,224,224
62323 DATA224,224,224,224,224,255,255,255
62330 DATA111,238,159,235,183,239,253,239
62331 DATA,,,,,128,192,224
62332 DATA7,3,1,,,,,
62333 DATA7,7,15,31,63,255,255,255
```



```
62334 DATA224 , 192 , 128 , , , , ,
62335 DATA255 , 255 , 255 , 7 , 7 , 135 , 139 , 231
62373 DATA , , , , 255 , 255 , 255 , 255
62408 DATA1
63000 FORI = 32821T032962 : READA : POKEI , A : NEXT
63001 DATA234 , 234 , 234 , 234 , 202 , 202 , 202 , 202, 96 , 234 , 234 , 234 , 234 , 232 , 232, 232 , 232
63002 DATA96 , 173 , 249 , 7, 201 , 13 , 240 , 20, 201 , 15 , 240 , 28 , 201 , 14 , 240 , 36 , 174 , 3, 208
63003 DATA32 , 62 , 128 , 142 , 3 , 208 , 76 , 131 , 128 , 174 , 2 , 208 , 32 , 53 , 128 , 142 , 2 , 208 , 76, 131
63004 DATA128 , 174 , 3 , 208 , 32 , 53 , 128 , 142 , 3 , 208 , 76 , 131 , 128 , 174 , 2 , 208 , 32 , 62 , 128
63005 DATA142 , 2 , 208 , 234 , 234 , 284 , 176 , 250 , 7 , 201 , 13 ,240 , 20 , 201 , 15 , 240 , 28 , 201 , 14
62006 DATA240 , 38 , 174 , 5 , 209 , 32 , 62 , 128 , 142 , 5 , 208 , 76 , 194 , 128 , 174 , 4 , 208 , 32 , 53 , 128
63007 DATA142 , 1 , 208 , 76 , 194 , 128 , 174 , 5 , 208 , 32 , 53 , 128 , 142 , 5 , 208 , 76 , 194 , 128 , 174
63008 DATA4 , 208 , 32 , 62 , 128,142 , 4 , 208 , 96
63020 FORI = 36864TO36974 : READA : POKEI , A : NEXT : RETURN
63021 DATA163 , 5 , 139 , 36 , 5 , 201 , 57 , 240 , 4 , 254 , 36 , 5 , 36 , 169 , 48 , 157 , 96 , 5 , 202 , 76 , 2
63022 DATA144 , 162 , 3 , 139 , 40 , 8 , 201 , 48 , 208 , 9 , 224 , 0 , 240 , 4 , 202 , 76 , 24 , 144 , 96 , 162
63023 DATA3 , 189 , 40 , 8 , 201 , 48 , 240 , 4 , 222 , 40 , 6 , 96 , 169 , 57 , 157 , 40 , 6 , 202 , 76 , 42 , 144
63024 DATA32 , 9 , 144 , 78 , 74 , 144 , 32 , 0 , 144 , 32 , 22 , 144 , 162 , 0 , 169 , 0 , 157 , 0 , 212 , 224 , 24
63025 DATA240 , 4 , 232 , 76 , 78 , 144 , 160 , 11 , 141 , 24 , 212 , 141 , 0 , 212 , 141 , 1 , 212 , 169 , 4
63026 DATA141, 5 , 212 , 169 , 33 , 141 , 4 , 212 , 96

READY.
```

Noah 2099

Noha - 2099

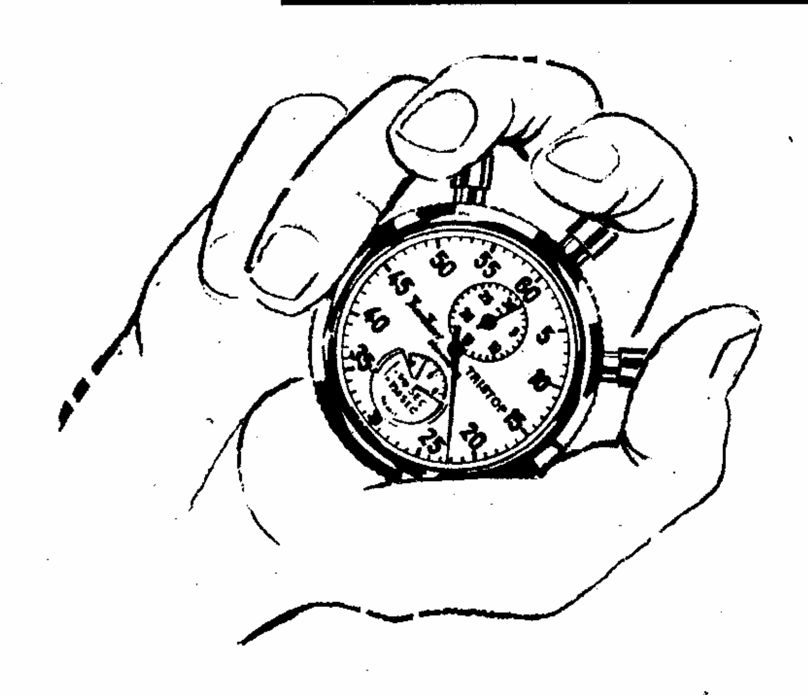
Ein Spiel in Extended Basic mit Joystick

Wir schreiben das Sternenjahr 2099. Im Sonnensystem XOR bahnt sich eine kosmische Katastrophe an, denn die Sonne bläht sich auf und schickt radioaktive Strahlung ins All, welche sämtliche Planeten des Sonnensystems XOR zu vernichten droht. Sie haben nun als NOHA die Aufgabe, sämtliche Lebewesen auf den Planeten zu evakuieren, damit diese nicht durch die Strahlung getötet werden. Steigen Sie nun ein in Ihre TI 99/4 A Arche, schieben Sie das Extended BASIC Modul in den Modulschacht und nehmen Sie den Joystick zur Hand. Es beginnt eine abenteuerliche Rettungsaktion im Sonnensystem XOR.

Das Spiel wechselt ständig zwischen zwei Spielszenen:

In der ersten Spielszene (Bild 1) fliegen Sie durch einen Meteoritenschwarm, dem es gilt auszuweichen. Steuern Sie Ihre Arche mit dem Joystick nach rechts oder links sicher durch die Meteoriten hindurch. Die Arche fliegt langsam von unten nach oben über den Bildschirm, wobei sich die Steuerung auf den Meteoritenschwarm auswirkt. Wenn die Arche den oberen Bildschirmrand erreicht hat, so wechselt das Bild blitzschnell zu Szene 2 (Bild 2) um. Die Arche senkt sich jetzt auf den Planeten herunter und verharrt etwa in Mitte des Bildschirms. Steuern Sie jetzt den Lander mittels des Joysticks aus der Arche und fliegen Sie zur Planetenoberfläche hinab, wo ein Wesen aufgeregt hin- und herläuft. Steuern Sie den Lander über das Wesen und drükken Sie den Aktionsknopf, um den Trägerstrahl auszulösen und das Wesen an Bord zu „beamen". Wenn Sie das geschafft haben, so kehren Sie mit dem Lander zur Arche zurück, wonach sich diese wieder vom Planeten entfernt und in den nächsten schnelleren Meteoritensturm hineinfliegt.

Es gibt manche Planeten, deren Athmosphäre so dicht ist, daß der Lander nicht zur Oberfläche vordringen kann.



In diesen Fällen muß die Atmosphäre mittels des ‚Beamstrahles' vernichtet werden, damit der Lander das Wesen retten kann.

Wenn der Lander nicht weit genug zur Oberfläche vordringt, so drücken Sie den Aktionsknopf. Sie sehen nun den ‚Beamstrahl'. Lassen Sie den Knopf gedrückt und bewegen Sie den Joystick nach oben, so lange, bis der ‚Beamstrahl' nicht mehr erscheint. Jetzt lassen Sie den Aktionsknopf los und drücken anschließend den Joystick nach unten. Die Atmosphäre ist jetzt dünner geworden und der Lander kann ungehindert zur Planetenoberfläche vordringen.

Für das Rettungsmanöver haben Sie selbstverständlich nur begrenzt Zeit. Wenn die Strahlung dem Planeten zu nahe kommt, so stößt die Arche ein Warnsignal aus. Jetzt heißt es, sich mit der Rettung zu beeilen, da der Lander nur nach erfolgreicher Mission zur Arche zurückkehren kann.

Das Spiel beginnt mit Runde 1 und hat seinen höchsten Schwierigkeitsgrad bei Runde 20 erreicht. Danach geht das Spiel zwar weiter, steigert jedoch die Schwierigkeit nicht mehr.

Für jeden bestandenen Meteoritensturm erhalten Sie 100mal Rundenzahl Punkte.

Jedes gerettete Wesen bringt 1000 plus 50mal Rundenzahl Punkte.

Alle 10 000 Punkte gibt es eine Bonus-Arche. Der Rechner kann maximal 15 Reservearchen bereithalten. Sie beginnen das Spiel mit 5 Archen. Die Punktanzeige geht bis 9 999 999. Danach werden Sternchen angezeigt.

Ich bin persönlich noch nicht in den Genuß gekommen, eine sechsstellige Zahl in der Punktanzeige zu sehen sowie mehr als sechs Archen mein eigen nennen zu dürfen.

Das Programm zeigt eine ungewöhnliche Programmtechnik der Sprites. So werden diese zum Teil derart überlagert, daß die Arche in zwei Farben erscheint. Weiterhin wurde sich der Unart des TI, nur max. 4 Sprites pro Zeile darstellen zu können, zu Nutze gemacht und somit das interessante ‚Verschwinden' der Arche realisiert. Soundmäßig wurde aus dem TI alles herausgeholt, was er hergibt.

Doch nun: Viel Spaß beim Spiel!

Anmerkung: Während des Spieles darf die Taste ‚ALPHA LOCK' nicht gedrückt sein, da sonst die Joysticksteuerung nicht richtig funktioniert.

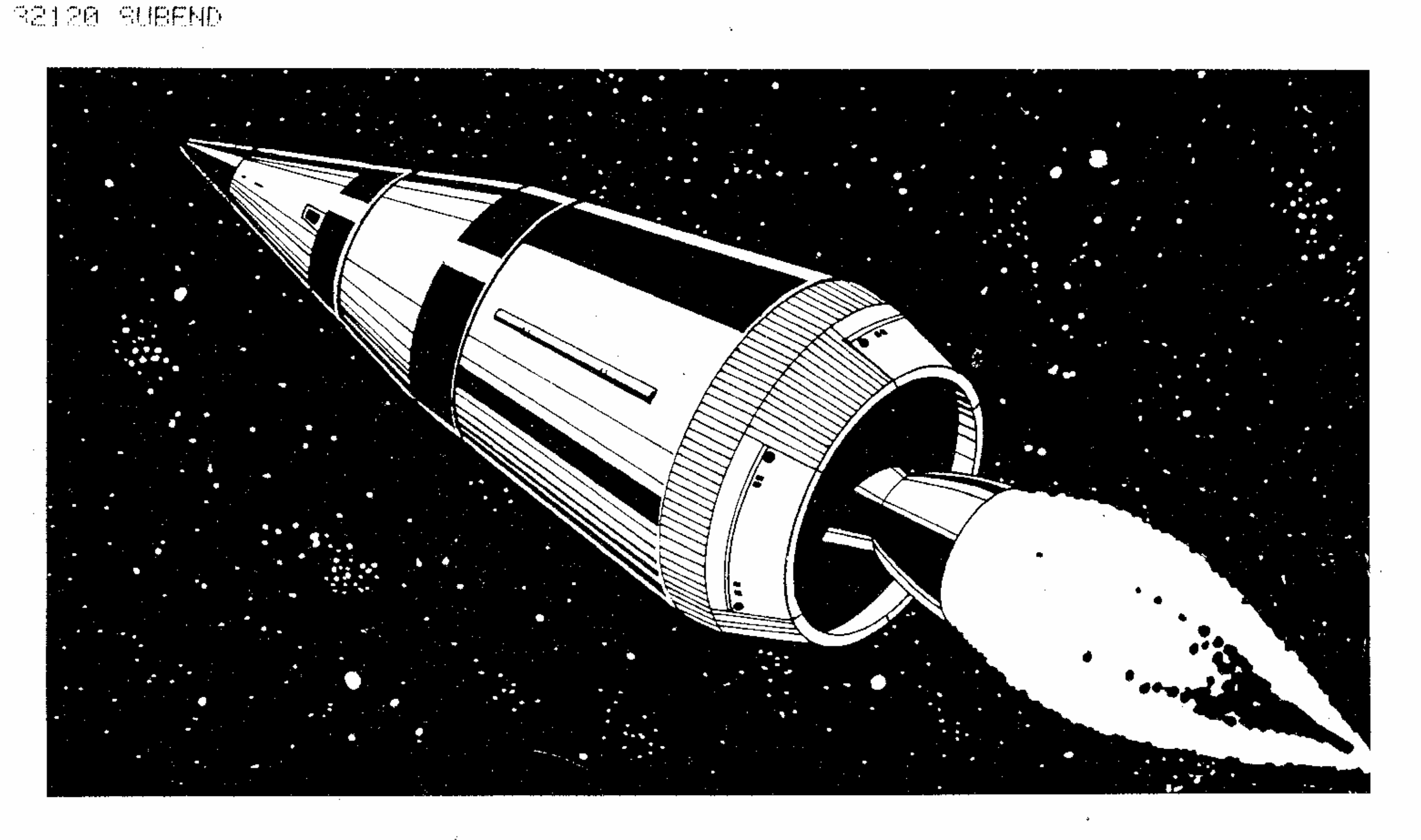
```
1 !@P-
10 REM ************************************
12 REM *          N O H A  -  2 0 9 9      *
14 REM *-----------------------------------*
16 REM * Ein Spiel fuer den TI 99/4 A      *
18 REM * geschrieben in  Extended BASIC   *
20 REM * Steuerung mittels Joystick        *
22 REM *-----------------------------------*
24 REM * (c) 1984 by Volker Becker        *
26 REM *      Steinbacher Strasse 10       *
28 REM *      6370 Oberursel 6             *
30 REM ************************************
32 REM
34 REM
70 !@P+
72 GOTO 90 :: ARCHE :: BONUS :: I :: K :: M :: MOT :: P :: R :: RUNDE :: S :: ST
 :: STV :: T :: X :: X1 :: YY :: Y :: Y1 :: CALL ANZARCHE :: CALL ASCIISET :: CA
LL CLEAR :: CALL COINC
74 CALL COLOR :: CALL DELSPRITE :: CALL HCHAR :: CALL JOYST :: CALL KEY :: CALL
MAGNIFY :: CALL MOTION :: CALL POSITION :: CALL PUNKTE :: CALL SCREEN
76 CALL SOUND :: CALL SPRITE :: CALL STRAHLUNG :: CALL TITEL
80 DATA 1,2,3,4,5,6,7,8,9,0
90 OPTION BASE 1
95 DIM B(4)
96 !@P-
100 RANDOMIZE
105 CALL CLEAR :: CALL ASCIISET :: CALL TITEL(B(),5,0):: CALL COLOR(2,10,1,3,10,
1,4,10,1)
108 CALL SCREEN(2):: CALL COLOR(1,11,1,12,10,1):: CALL CLEAR
109 FOR K=33 TO 124 STEP 91
110 FOR I=1 TO 50 :: X=RND*21+1 :: Y=RND*31+1 :: CALL HCHAR(X,Y,K):: NEXT I
112 NEXT K
120 RUNDE=1 :: ARCHE=4 :: P=0 :: BONUS=1
124 CALL ANZARCHE(ARCHE):: CALL PUNKTE(P)
125 CALL MAGNIFY(4)
130 CALL SPRITE(#1,128,9,193,96,#2,132,9,193,128)
140 CALL MOTION(#1,-3,0,#2,-3,0)
150 X=RND*139+1 :: Y=X+115 :: CALL SPRITE(#10,96,4,1,X,#11,100,14,57,Y)
160 X=RND*139+1 :: Y=X+115 :: CALL SPRITE(#12,104,11,121,X,#13,108,8,185,Y)
162 MOT=10+RUNDE
165 FOR I=10 TO 13 :: CALL MOTION(#I,MOT,5):: NEXT I
170 CALL JOYST(1,X,Y)
175 IF X=0 THEN 190
180 FOR I=10 TO 13 :: CALL MOTION(#I,MOT,-X*4):: NEXT I
190 CALL COINC(ALL,ST):: IF ST THEN GOSUB 2000 :: GOTO 130
200 CALL POSITION(#1,X,Y):: IF X<7 THEN 1000
210 CALL SOUND(-500,520-X,5,518-X,5,522-X,5,-7,5):: GOTO 170
1000 P=P+100*RUNDE :: CALL PUNKTE(P):: GOSUB 8000
1005 CALL MOTION(#1,0,0,#2,0,0):: FOR I=10 TO 13 :: CALL DELSPRITE(#I):: NEXT I
1010 CALL SCREEN(5):: CALL COLOR(1,4,1):: CALL COLOR(#1,11,#2,11):: STV=0
1020 CALL HCHAR(24,1,35,32):: CALL HCHAR(23,1,34,32)
1022 CALL POSITION(#1,X,Y):: CALL SPRITE(#3,136,9,X,Y,#4,140,9,X,Y+32)
1025 CALL MOTION(#1,3,0,#2,3,0,#3,3,0,#4,3,0)
1027 CALL SPRITE(#5,112,2,147,112)
1030 FOR I=1 TO 10
1035 MOT=-MOT :: CALL MOTION(#5,0,MOT)
1050 CALL SOUND(150,195,0):: CALL SOUND(150,261,0):: CALL SOUND(150,329,0):: CAL
L SOUND(150,523,0)
1060 NEXT I
1070 CALL MOTION(#1,0,0,#2,0,0):: CALL DELSPRITE(#3,#4)
```

```
1080 CALL POSITION(#1,XV,Y):: CALL SPRITE(#3,116,14,XV+5,Y+16)
1090 FOR I=90-RUNDE*3 TO 0 STEP -1
1092 CALL COINC(ALL,ST):: IF ST THEN 4000
1095 M=RND*20 :: IF M>10 THEN MOT=-MOT
1100 CALL JOYST(1,X,Y):: CALL POSITION(#3,X1,Y1):: IF STV THEN 1102 ELSE IF X1<=
XV+32 THEN IF Y=4 THEN Y=0
1102 IF X1>=125 THEN IF Y=-4 THEN Y=0
1105 CALL MOTION(#3,-Y*5,X*5,#5,0,MOT):: CALL KEY(1,T,S):: IF T=18 THEN GOSUB 30
00
1107 IF I<=15 THEN CALL SOUND(-100,195,0,261,0,329,0,-7,0)ELSE CALL SOUND(-100,6
59,0)
1110 NEXT I
1112 CALL COLOR(#1,6,#2,6)
1115 CALL STRAHLUNG
1120 GOSUB 2000
1130 CALL HCHAR(23,1,32,64):: CALL SCREEN(2):: CALL COLOR(1,11,2):: GOTO 130
2000 CALL MOTION(#1,0,0,#2,0,0):: CALL DELSPRITE(#3,#5):: CALL POSITION(#1,X,Y)
2005 FOR I=10 TO 13 :: CALL DELSPRITE(#I):: NEXT I
2010 CALL SPRITE(#12,128,6,X,Y,#13,132,6,X,Y+32):: CALL SPRITE(#10,128,1,X,Y,#11
,132,1,X,Y+32)
2020 CALL SPRITE(#1,128,1,X+32,Y,0,0,#2,132,1,X+32,Y+32)
2030 CALL MOTION(#1,-5,0,#2,-5,0)
2040 FOR I=523 TO 343 STEP -10
2050 CALL SOUND(-50,I,0,I+1,0,I-1,0)
2060 NEXT I
2070 CALL DELSPRITE(ALL)
2075 ARCHE=ARCHE-1 :: CALL ANZARCHE(ARCHE):: IF ARCHE<0 THEN 10000
2080 RETURN
3000 CALL MOTION(#3,0,0):: CALL POSITION(#3,X,Y):: IF X<125 THEN RETURN ELSE CAL
L SPRITE(#4,120,12,X,Y):: CALL SOUND(-250,1046,0)
3010 CALL COINC(ALL,ST):: IF ST THEN 5000
3015 CALL DELSPRITE(#4)
3020 RETURN
4000 IF STV THEN 6000 ELSE 1095
5000 STV=-1 :: CALL DELSPRITE(#4,#5):: CALL SOUND(250,523,0):: GOTO 3020
6000 P=P+1000+50*RUNDE :: CALL PUNKTE(P):: GOSUB 8000
6005 CALL DELSPRITE(#3,#4):: CALL POSITION(#1,X,Y):: CALL SPRITE(#3,136,9,X,Y,#4
,140,9,X,Y+32)
6010 CALL MOTION(#1,-4,0,#2,-4,0,#3,-4,0,#4,-4,0)
6015 RUNDE=RUNDE+1 :: IF RUNDE>20 THEN RUNDE=20
6020 FOR I=1 TO 10
6030 CALL SOUND(150,523,0):: CALL SOUND(150,329,0):: CALL SOUND(150,261,0):: CAL
L SOUND(150,195,0)
6040 NEXT I
6050 CALL DELSPRITE(ALL)
6060 GOTO 1130
8000 IF P>=10000*BONUS THEN 8020
8010 RETURN
8020 ARCHE=ARCHE+1 :: IF ARCHE>15 THEN ARCHE=15
8030 CALL ANZARCHE(ARCHE)
8035 BONUS=BONUS+1
8040 RETURN
10000 CALL TITEL(B(),1,1)
10010 GOTO 108
19999 !@P+
20000 SUB EINLEITUNG(B(),KY)
20002 GOTO 20010 :: S :: T :: CALL OLMANRIVER :: !@P-
20010 CALL DELSPRITE(ALL):: CALL CLEAR :: CALL COLOR(2,2,1):: CALL SCREEN(3)
20020 DISPLAY AT(1,1):"      N O H A - 2 0 9 9          ((((((((((((((((("
20030 DISPLAY AT(4,1):"IM SONNENSYSTEM XOR HAT SICHEINE KOSMISCHE KATASTROPHE  E
```

```
REIGNET. DIE SONNE BLAEHT  SICH AUF UND SCHICKT RADIO-"
20040 DISPLAY AT(8,1):"AKTIVE STRAHLUNG INS ALL."
20050 DISPLAY AT(10,1):"WIR SCHREIBEN DAS JAHR 2099 SIE HABEN NUN ALS NOHA DIE
AUFGABE DIE PLANETEN DES     SONNENSYSTEMES ZU EVAKUIEREN"
20060 DISPLAY AT(14,1):"BEVOR DIE STRAHLUNG DIESE   ERREICHT, UND SOMIT SAEMT-
LICHES LEBEN AUF DEN         PLANETEN VERNICHTET."
20070 DISPLAY AT(19,1):"FLIEGEN SIE MIT IHRER ARCHE VON PLANET ZU PLANET UND
RETTEN SIE DIE WESEN, DIE    AUF DEN PLANETEN LEBEN."
20080 DISPLAY AT(24,1):"WEITER MIT  E N T E R"
20085 CALL OLMANRIVER(B(),5)
20090 CALL KEY(5,T,S):: IF S<>0 AND T=13 THEN 20100 ELSE 20085
20100 CALL CLEAR
20110 DISPLAY AT(1,1):"STEUERN SIE IHRE ARCHE MIT  DEM JOYSTICK SICHER DURCH   E
INEN METEORITENSTURM, UM   DANN EINEN PLANETEN ZU"
20120 DISPLAY AT(5,1):"ERREICHEN. DANACH STEUERN   SIE DEN LANDER AUF DIE OBER-F
LAECHE DES PLANETEN HINAB  UND BEAMEN DAS WESEN IN DEN"
20130 DISPLAY AT(9,1):"LANDER. NUN KEHREN SIE MIT  DEM LANDER ZUR ARCHE ZURUECKU
ND FLIEGEN DURCH EINEN      WEITEREN METEORITENSTURM"
20140 DISPLAY AT(13,1):"ZUM NAECHSTEN PLANETEN.     DIE METEORITENSTUERME WERDEN
IMMER SCHNELLER. DIE ZEIT,  IN WELCHER SIE DAS WESEN AN"
20150 DISPLAY AT(17,1):"BORD DES LANDERS BEAMEN     MUESSEN WIRD IMMER KUERZER
UND DAS WESEN LAEUFT IMMER  SCHNELLER HIN UND HER."
20160 DISPLAY AT(21,1):"WENN DAS WARNSIGNAL DER     ARCHE ERTOENT, MUESSEN SIE
SICH MIT DER RETTUNG BEEILENWEITER MIT  E N T E R"
20165 CALL OLMANRIVER(B(),5)
20170 CALL KEY(5,T,S):: IF S<>0 AND T=13 THEN 20180 ELSE 20165
20180 CALL CLEAR
20190 DISPLAY AT(1,1):"SIE KOENNEN NUR ZUR ARCHE   ZURUECKKEHREN, WENN SIE DAS W
ESEN GERETTET HABEN."
20200 DISPLAY AT(5,1):"STEUERUNG                   ((((((((                   D
IE ARCHE, SOWIE DER LANDER WERDEN MIT DEM JOYSTICK GE- STEUEUERT."
20210 DISPLAY AT(10,1):"ZUM AUSLOESEN DES TRAEGER-  STRAHLES WIRD DER AKTIONS-
KNOPF GEDRUECKT."
20220 DISPLAY AT(14,1):"PUNKTWERTUNG                ((((((((((((
BESTANDENER METEORITENSTURM RUNDE MAL 100 PUNKTE"
20230 DISPLAY AT(18,1):"GERETTETES WESEN            1000 + RUNDE MAL 50 PUNKTE"
20240 DISPLAY AT(21,1):"ZU SPIELBEGINN STEHEN IHNEN 5 ARCHEN ZUR VERFUEGUNG.
BONUS ARCHE ALLE 10000 PUNKESPIELBEGINN MIT AKTIONSKNOPF"
20245 CALL OLMANRIVER(B(),1)
20250 CALL KEY(1,T,S):: IF T<>18 THEN 20245
20260 CALL CLEAR
20265 !@P+
20270 SUBEND
25000 SUB OLMANRIVER(B(),KY)
25005 GOTO 25010 :: AZ :: I :: J :: K :: M1 :: M2 :: S :: STV :: T :: !@P-
25010 DATA 1,195,261,329,391,4,329,391,1,174,220,261,349,1,261,440,2,261,523,1,2
61,440,1,195,261,329,391
25020 DATA 4,329,391,1,174,220,261,349,1,261,440,2,261,523,1,261,587,1,195,261,3
29,391,4,391,659
25030 DATA 1,174,220,261,349,1,349,587,2,349,523,1,349,587,1,195,261,329,391,2,3
91,659,2,391,783
25040 DATA 1,174,220,261,349,1,523,880,2,523,783,1,523,880,4,195,246,293,391,4,4
93,787
25050 DATA 1,493,659,2,493,587,1,493,659,4,493,783,1,493,659,2,493,587,1,493,659
25060 DATA 1,195,261,329,391,4,329,523,1,174,220,261,349,1,261,440,2,261,391,1,2
61,440
25070 DATA 2,195,261,329,391,8,329,523,0
25080 RESTORE 25010
25090 AZ,STV=0
25100 READ K :: IF K=0 THEN 25080
```

```
25110 READ B(1),B(2),B(3),B(4)
25120 FOR I=1 TO K
25130 FOR J=1 TO 4
25140 STV=STV+1 :: IF STV>AZ THEN READ AZ,M1,M2 :: STV=1
25150 CALL SOUND(400,M1,1,M2,0,B(J),5):: CALL KEY(KY,T,S):: IF S<>0 THEN SUBEXIT
25160 NEXT J :: NEXT I
25170 GOTO 25100
25175 !@P+
25180 SUBEND
28000 SUB STRAHLUNG
28005 GOTO 28010 :: C :: I :: K :: !@P-
28010 DATA 11,12,16
28020 RESTORE 28010
28030 FOR K=3 TO 1 STEP -1
28035 READ C :: CALL SCREEN(C)
28040 FOR I=10 TO 1 STEP -1
28050 CALL SOUND(-250,-7,K*I)
28060 NEXT I
28070 NEXT K
28075 !@P+
28080 SUBEND
30000 SUB ANZARCHE(ARCHE)
30005 !@P-
30010 CALL HCHAR(1,3,32,15)
30020 IF ARCHE<=0 THEN SUBEXIT
30030 CALL HCHAR(1,3,40,ARCHE)
30035 !@P+
30040 SUBEND
30050 SUB PUNKTE(P)
30055 !@P-
30060 DISPLAY AT(1,22):USING "#######":P
30065 !@P+
30070 SUBEND
31000 SUB TITEL(B(),KY,STN)
31001 GOTO 31002 :: S$ :: AS :: I :: J :: L :: S :: T :: CALL EINLEITUNG :: CALL
 VCHAR :: !@P-
31002 CALL MAGNIFY(2)
31005 IF STN=1 THEN RESTORE 31035 :: CALL SCREEN(5):: CALL COLOR(1,11,5):: GOTO
31055
31010 CALL CLEAR :: CALL SCREEN(5):: CALL COLOR(2,11,1)
31020 CALL HCHAR(1,3,40,28):: CALL HCHAR(2,3,40,28):: CALL HCHAR(23,3,40,28):: C
ALL HCHAR(24,3,40,28)
31030 CALL VCHAR(3,3,40,20):: CALL VCHAR(3,4,40,20):: CALL VCHAR(3,29,40,20):: C
ALL VCHAR(3,30,40,20)
31035 DATA G,A,M,E,O,V,E,R
31040 DATA N,O,H,A,2,0,9,9
31045 DATA 128,130,132,134,129,131,133,135
31050 RESTORE 31040
31055 J=0
31060 FOR I=73 TO 169 STEP 32
31065 J=J+1
31070 READ S$ :: CALL SPRITE(#J,ASC(S$),4,40,I)
31080 NEXT I
31090 FOR I=73 TO 169 STEP 32
31100 J=J+1
31110 READ S$ :: CALL SPRITE(#J,ASC(S$),9,89,I)
31120 NEXT I
31121 IF STN=1 THEN 31125
31122 CALL COLOR(13,16,1)
```

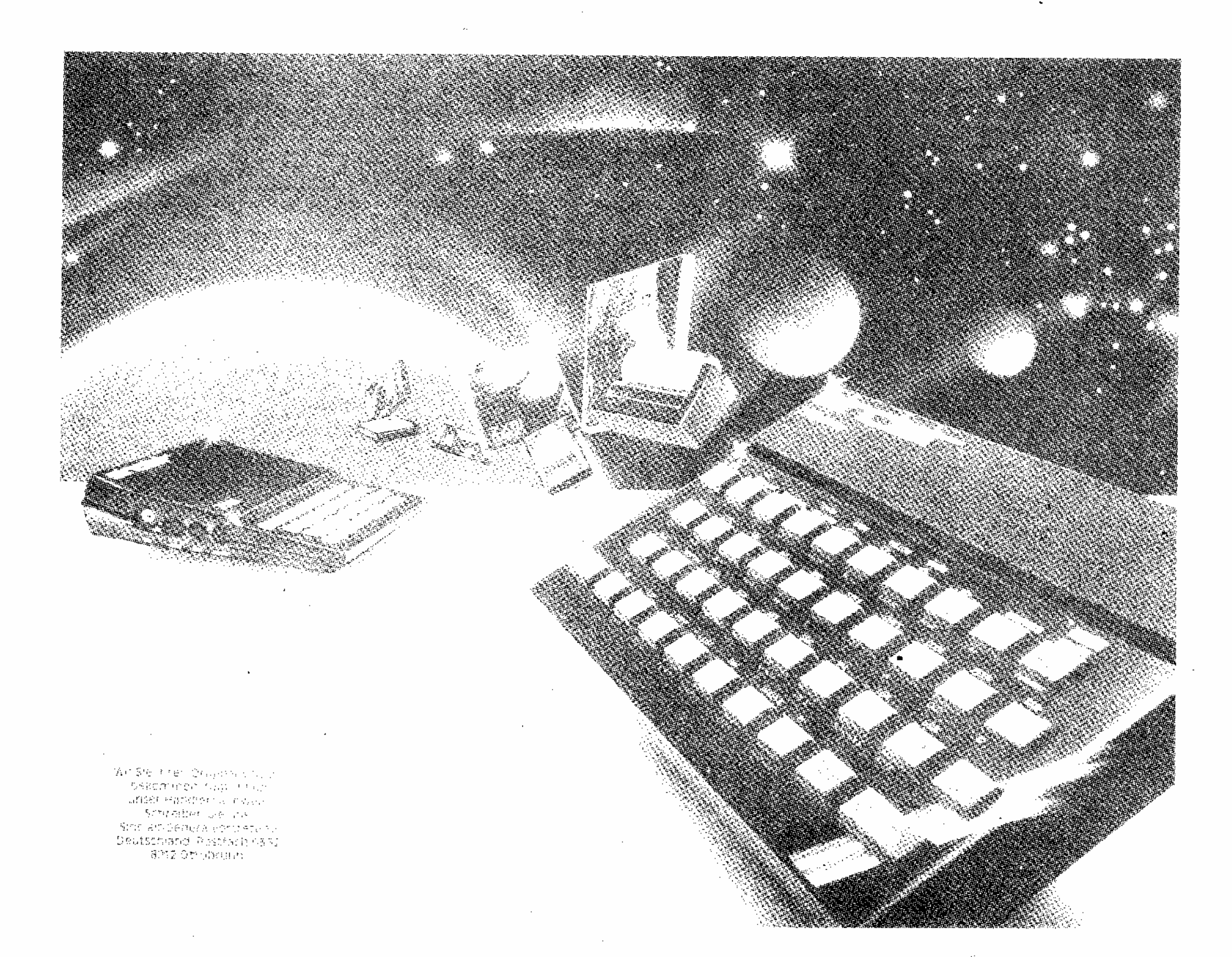
```
31123 FOR L=9 TO 10 :: FOR S=15 TO 18 :: READ AS :: CALL HCHAR(L,S,AS):: NEXT S
:: NEXT L
31124 DISPLAY AT(18,4)SIZE(20):"SPIELANLEITUNG <J/N>"
31125 DISPLAY AT(21,4)SIZE(23):"@1984 BY VOLKER BECKER"
31130 CALL OLMANRIVER(B(),KY)
31135 IF STN=0 THEN 31160
31140 CALL CLEAR :: CALL DELSPRITE(ALL)
31150 SUBEXIT
31160 CALL KEY(5,T,S):: IF T<>74 THEN IF T<>106 THEN 31140
31170 CALL EINLEITUNG(B(),KY)
31175 !@P+
31180 SUBEND
32000 SUB ASCIISET
32005 GOTO 32010 :: C$ :: C :: I :: CALL CHAR :: !@P-
32010 DATA 00000000000010306E,070301,000000000080C0E0,B060C08,000000000010307,0D0
60301,000000008080C06,E0C08,000000000103060D,070301
32020 DATA 0000000000080C0E,70C08,000000000010307,06030101,0000000080C060B,E0C08
,,0103050306 0F0206,,80C0A0C060F0406,000000001071F0A
32030 DATA 1F01,000000080E0F85,F88,,0000000101010101,,0000008080808 08,000000281
028,,,,000001071F7FFF66,FF7F1F0701,0F3FE30FE3FFFF66,FFFFE3F1F83F0F03
32040 DATA F0FCC7F0C7FFFF66,FFFFC78F1FFCF0C,000080E0F8FEFF66,FFFEF8E08,000000010
0003F3F,3F,003E3EF83F00FFFF,FF3E3E1F0F,007C7C1F7C00FFFF
32050 DATA FF7C7CF8F,0000000800000FCFC,FC,33,000000281028,34,183C77FDFFB7FDEF,35,
F7FFFDFFBBFFFFF,40,00187E247E18,64,3C4299A1A199423C
32060 DATA 003C666666666663C,001838181818183C,003C66060C18307E,003C66061C06663C,0
00C1C2C4C7E0C0C,007E607C0606663C,001E30607C66663C
32070 DATA 007E060C1830303,003C66663C66663C,003C66663E060C38
32080 RESTORE 32010
32090 FOR I=96 TO 143 :: READ C$ :: CALL CHAR(I,C$):: NEXT I
32100 FOR I=1 TO 5 :: READ C,C$ :: CALL CHAR(C,C$):: NEXT I
32110 FOR I=48 TO 57 :: READ C$ :: CALL CHAR(I,C$):: NEXT I
32115 !@P+
32120 SUBEND
```



SEIKOSHA Grafik-Drucker

GP-50A und GP-50S DER KLEINE LOW-COST-DRUCKER

- Normalpapier (Einzelblatt + Rolle)
- 40 Zeichen/Sek.
- 46 Zeichen/Zeile
- 5 x 8 Punktmatrix
- Grafik durch Einzelpunkt-Ansteuerung
- Option: 6 verschiedene Farbband-Kassetten (rot, orange, grün, blau, violett und braun)
- Direkt an Sinclair ZX-81 oder Spectrum anschließbar (GP-50S)
- Schnittstelle: Centronics-kompatibel (GP-50A)



Jack the Digger

JACK THE DIGGER

Das Programm ist in Extended-Basic geschrieben und benötigt Joysticks

Jack's Aufgabe besteht darin, Glückssymbole aufzusammeln, die in den Stollen seiner Mine verteilt sind.

Um ein Symbol aufzunehmen, muß Jack nur darübergehen. Jedes Symbol hat einen bestimmten Wert, der vom Computer zufällig ausgewählt wird.

Haben die Symbole einen besonders hohen Wert (über 50 Punkte), wird bei deren Verteilung in den Stollen ein Tusch gespielt. Um einen besonders hohen High-Score zu bekommen, sollten gerade diese Symbole alle aufgesammelt werden. Es müssen nicht alle Symbole aufgesammelt werden. Wenn keine Möglichkeit mehr besteht, an Symbole zu kommen, dann muß Jack über die letzte Leiter in den untersten Stollen gehen. Bei dem Wort „Lift" ist eine Glocke dargestellt. Wenn Jack diese Glocke berührt, setzt sich ein Lift in Bewegung, der ihn dann in den ersten Stollen nach oben bringt, wo er seine gesammelten Punkte abliefern kann.

Für jede Runde wird dann im obersten Stollen ein Sack dargestellt. Das Spiel ist zu Ende, wenn 10 Säcke erfolgreich nach oben gebracht wurden oder wenn Jack alle Leben verloren hat.

Die Aufgabe von Jack wird dadurch erschwert, daß in den Stollen böse Geister leben. Sie bewachen die Glückssymbole. Von den Geistern darf sich Jack nicht erwischen lassen. Ebenso tödlich für ihn ist ein Berühren der Stollenwände oder ein Fehltritt auf einer Leiter. Anfang und Ende eines jeden Stollens bieten in den ersten Runden eine relative Sicherheit vor den Geistern. Absolut sicher ist Jack auf den Leitern. Sie bieten den sichersten Schutz.

Pro Runde werden die Geister schneller. Dann gilt es manchmal nur noch, das Leben von Jack zu retten (mit einem leeren Sack nach oben fahren). Verliert Jack sein Leben, muß er zur Strafe wieder von ganz oben anfangen.

Noch ein Hinweis zum Abschreiben des Programms.

Manche Zeilen werden wegen ihrer Länge nicht ganz angenommen. In diesem Fall beenden Sie die Eingabe mit „Enter". Rufen Sie diese Zeile durch gleichzeitiges Drücken von „FTCN + REDO" wieder zurück und schreiben Sie die Zeile dann komplett fertig.

Die Graphik der Mine wird mit „FOR-NEXT"-Schleifen aufgebaut. Mit „DISPLAY AT" wäre der Aufbau schneller. Da aber die Zeichen für die Stollenwände im frei definierbaren Bereich liegen, wäre die Darstellung im Listing erschwert worden.

Und nun viel Spaß und Punkte bei diesem Spiel.

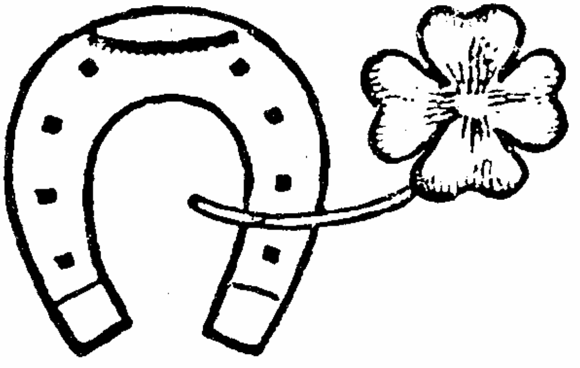


****

```
100 ! JACK THE DIGGER
110 ! WRITTEN BY W.DOELTSCH
120 ! 6108 WEITERSTADT 3
130 ! RAPPMUEHLSTR.58 C
140 ! TEL. 06150-40547
150 CALL CLEAR :: CALL MAGNIFY(2):: I=3 :: FOR T=1 TO 16 :: READ A :: CALL SPRIT
E(#T,A,5,I*16+2,I+190,0,-19):: I=I+1 :: IF A=32 THEN I=3
160 NEXT T
170 DATA 74,65,67,75,32,84,72,69,32,68,73,71,71,69,82,32
180 RANDOMIZE :: CALL SCREEN(12):: DISPLAY AT(12,3)BEEP:"ERKLAERUNGEN ??(J oder
N)"
190 CALL KEY(2,T,S):: IF T=15 THEN 230 ELSE IF T=2 THEN 200 ELSE 190
200 CALL CLEAR :: CALL SCREEN(13):: FOR I=1 TO 12 :: CALL COLOR(I,5,12):: NEXT I
210 DISPLAY AT(3,1):"***** JACK the DIGGER *****": :"Sammle alle Symbole aber":
:"beruehre nicht die Wand": :"oder die Geister."
220 DISPLAY AT(13,1):"Bringe die Punkte zum Lift": : :"10 Runden gibt's": :"Pro
Runde wird's schwerer." :: CALL TASTER
230 DISPLAY AT(10,1)BEEP ERASE ALL:"ALPHA LOCK TASTE NACH OBEN!!" :: CALL TASTER
 :: CALL SCREEN(2):: CALL COLOR(1,2,2):: FOR I=2 TO 11 :: CALL COLOR(I,16,2):: N
EXT I
```

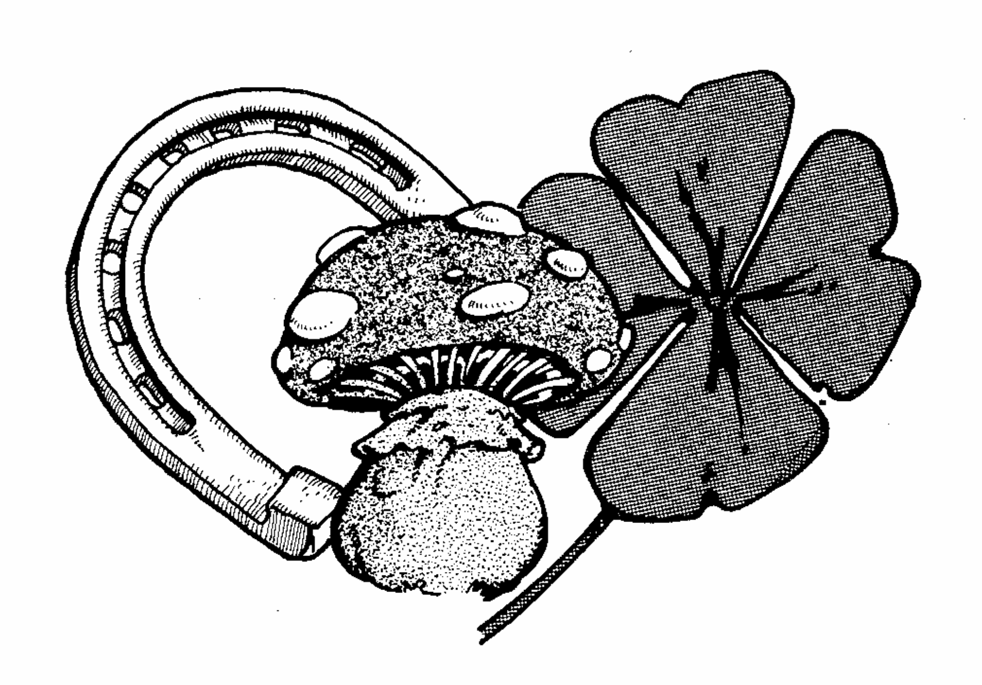
```
240 CALL FF :: DIM M(18):: NU,TOT=0 :: EX=232 :: DEF FA=INT(RND*(14-3+1))+3
250 CALL CHAR(128,"E0F8F0FEFCF0C0E0",129,"C0F8E0FCF8C0F0E0",130,"F0F8FCFEF0F8E0F
0",131,"E0F8F0FEF8FCF8E0")
260 CALL CHAR(132,"030F071F3F0F0703",133,"071F070F1F3F0F03",134,"070F3F03071F030
7",135,"03070F1F3F0F0307")
270 CALL CHAR(136,"FFFF7F1D08",137,"FFFEFE3C3C18",138,"FFFFFE7A70202",139,"FFFFF
E74501")
280 CALL CHAR(140,"0010185A7AFFFFFF",141,"0000081A1E5FFFFF",142,"000010185C7DFDF
F",143,"000000103476F6FF")
290 CALL CHAR(124,"C0F0E0E4FCFCFEFF",125,"0707131F4F5FFFFF",126,"FFFEF8FCE4F0E0C
0",127,"FFFF7F1F3F0B0307")
300 CALL CHAR(40,"81FF81FF81FF81FF",42,"81818181818181FF",106,"FFFFFFFFFFFFFFFF"
,86,"0000000000000018")
310 CALL CHAR(105,"6C38106C9F825C38",41,"10107C10101038FE"):: K1$="1C1E3E7CFCAC4
6BE" :: K2$="38787C3E3F35627D" :: CALL CHAR(43,K1$)
320 CALL CHAR(112,"303428303030486C",113,"3030203C30302030",114,"0C2C140C0C0C123
6",115,"0C0C043C0C0C040C")
330 CALL CHAR(116,"383810381428680C",117,"3838107C10282C60",118,"000000412112DCF
C",119,"0000001212D2DC3C")
340 CALL HCHAR(24,1,106,32):: CALL VCHAR(1,1,106,23):: CALL VCHAR(1,32,106,23)
350 FOR R=3 TO 21 STEP 3 :: FOR P1=5 TO 29 STEP 4 :: CALL HCHAR(R,P1,140):: NEXT
 P1 :: FOR P2=6 TO 30 STEP 4 :: CALL HCHAR(R,P2,141):: NEXT P2
360 FOR P3=7 TO 27 STEP 4 :: CALL HCHAR(R,P3,142):: NEXT P3 :: FOR P4=8 TO 28 ST
EP 4 :: CALL HCHAR(R,P4,143):: NEXT P4 :: NEXT R
370 FOR R=4 TO 19 STEP 3 :: FOR P1=5 TO 29 STEP 4 :: CALL HCHAR(R,P1,136):: NEXT
 P1 :: FOR P2=6 TO 30 STEP 4 :: CALL HCHAR(R,P2,137):: NEXT P2
380 FOR P3=7 TO 27 STEP 4 :: CALL HCHAR(R,P3,138):: NEXT P3 :: FOR P4=8 TO 28 ST
EP 4 :: CALL HCHAR(R,P4,139):: NEXT P4 :: NEXT R :: CALL HCHAR(22,5,32,27)
390 FOR R=2 TO 21 STEP 4 :: CALL VCHAR(R,2,128):: CALL VCHAR(R+1,2,129):: CALL V
CHAR(R+2,2,130):: CALL VCHAR(R+3,2,131):: NEXT R
400 FOR R=2 TO 21 STEP 4 :: CALL VCHAR(R,31,132):: CALL VCHAR(R+1,31,133):: CALL
 VCHAR(R+2,31,134):: CALL VCHAR(R+3,31,135):: NEXT R
410 FOR R=2 TO 29 STEP 4 :: CALL VCHAR(1,R,136):: CALL VCHAR(1,R+1,137):: CALL V
CHAR(1,R+2,138):: CALL VCHAR(1,R+3,139):: NEXT R :: CALL HCHAR(1,30,136,2)
420 FOR R=4 TO 16 STEP 3 :: CALL VCHAR(R,4,132):: CALL VCHAR(R+1,4,133):: CALL V
CHAR(R+2,4,134):: NEXT R
430 FOR R=5 TO 17 STEP 3 :: CALL VCHAR(R,5,129):: NEXT R :: FOR R=1 TO 19 STEP 3
 :: CALL VCHAR(R,31,127):: NEXT R
440 FOR R=4 TO 16 STEP 3 :: CALL VCHAR(R,5,126):: NEXT R :: FOR R=6 TO 19 STEP 3
 :: CALL VCHAR(R,5,124):: NEXT R
450 CALL HCHAR(1,2,126):: CALL HCHAR(21,2,124):: FOR R=3 TO 21 STEP 3 :: CALL VC
HAR(R,31,125):: NEXT R :: CALL HCHAR(21,3,143,2)
460 DISPLAY AT(22,1)SIZE(9):"POINTS :" :: DISPLAY AT(22,16)SIZE(6):"LIFES:" ::
CALL HCHAR(22,25,113,5)
470 DISPLAY AT(23,1)SIZE(9):"HI-SCORE:" :: DISPLAY AT(23,16)SIZE(6):"SCORE:" ::
DISPLAY AT(20,1)SIZE(5):"LIFE+"
480 CALL DELSPRITE(ALL) :: CALL MAGNIFY(1)
490 CALL LEER :: GOSUB 890 :: GOSUB 960
500 H=32 :: V=40 :: CALL DELSPRITE(ALL)
510 CALL SPRITE(#1,112,15,32,40,#7,42,5,8,EX,#28,32,1,16,232)
520 RANDOMIZE :: ON INT(RND*20)+1 GOTO 530,540,550,560,570,580,590,600,610,620,6
30,640,650,660,670,680,690,700,710,720
530 A$="0024247E427E4281" :: GOTO 730
540 A$="6699997E247EA5A5" :: GOTO 730
550 A$="81C36699997E2418" :: GOTO 730
560 A$="6699997E245A99A5" :: GOTO 730
570 A$="42E7999966660B81" :: GOTO 730
580 A$="187EFF9999FFDB99" :: GOTO 730
590 A$="003C7E99FF7E42C3" :: GOTO 730
600 A$="1C3E6B7F223E2277" :: GOTO 730
610 A$="92BAD67C6C3892FE" :: GOTO 730
620 A$="81423C5A7E1824E7" :: GOTO 730
630 A$="245A99BD3C5A6618" :: GOTO 730
```

```
640 A$="66993C5A247E81E7" :: GOTO 730
650 A$="FF99FF5A0000A5FF" :: GOTO 730
660 A$="3C7E99FFFFDBA5FF" :: GOTO 730
670 A$="183C563C52492A49" :: GOTO 730
680 A$="FF9999FFE7C3C381" :: GOTO 730
690 A$="3C7EFF99FF9981FF" :: GOTO 730
700 A$="006666003C66C381" :: GOTO 730
710 A$="A5FF995A3C5A8181" :: GOTO 730
720 A$="81A55A3C243C42C3"
730 CALL CHAR(96,A$):: CALL SPRITE(#2,96,FA,32,232,#3,96,FA,56,40,#4,96,FA,80,23
2,#5,96,FA,104,40,#6,96,FA,128,232):: GOTO 1200
740 CALL COINC(ALL,HIT):: IF HIT THEN 810 :: CALL POSITION(#28,X1,X2):: IF X2<60
 THEN 1230 ELSE IF X2>200 THEN 1220
750 CALL JOYST(2,X,Y):: IF X=0 AND Y=0 THEN 740 ELSE IF X=4 THEN CALL PATTERN(#1
,112):: V=V+8 :: PA=113 :: GOTO 780
760 IF X=-4 THEN CALL PATTERN(#1,114):: V=V-8 :: PA=115 :: GOTO 780 ELSE IF Y=4
THEN CALL PATTERN(#1,116):: H=H-8 :: PA=117 :: GOTO 780
770 IF Y=-4 THEN CALL PATTERN(#1,117):: H=H+8 :: PA=116
780 CALL COINC(ALL,HIT):: IF HIT THEN 810 :: CALL LOCATE(#1,H,V):: CALL PATTERN(
#1,PA)
790 CALL COINC(ALL,HIT):: IF HIT THEN 810 :: CALL GCHAR(H/8+1,V/8+1,C):: IF C=32
 THEN 740
800 IF C=88 THEN 880 ELSE IF C>127 THEN 810 ELSE IF C=43 THEN 1240 ELSE 740
810 TOT=TOT+1 :: FL=5 :: CALL DELSPRITE(#2,#3,#4,#5,#6,#28)
820 FL=FL+3 :: CALL SOUND(-200,330+(FL*10),0,335+(FL*10),0,340+(FL*10),0)
830 CALL POSITION(#1,ZS,SS):: IF ZS<135 THEN CALL MOTION(#1,FL,0):: GOTO 820 ELS
E CALL MOTION(#1,0,0)
840 CALL LOCATE(#1,154,SS):: CALL PATTERN(#1,118)
850 FOR I=1 TO 30 STEP 2.5 :: CALL SOUND(-500,-6,I):: CALL PATTERN(#1,119):: FOR
 D=1 TO 10 :: NEXT D :: CALL PATTERN(#1,118):: FOR D=1 TO 10 :: NEXT D :: NEXT I
860 CALL DELSPRITE(#1):: CALL HCHAR(22,25,41,TOT):: IF TOT=5 THEN 1600
870 H=32 :: V=40 :: GOSUB 940 :: GOSUB 890 :: GOTO 510
880 NU=NU+PU :: DISPLAY AT(22,10)SIZE(5):NU :: CALL HCHAR((H/8)+1,(V/8)+1,32)::
GOTO 740
890 !LEITERAUFBAU
900 D=1 :: PR=0 :: FOR L=6 TO 18 STEP 3
910 M(L)=INT((RND*((20-11+1))+11)+D)
920 IF (SGN(M(L)-PR)<>D)THEN 910
930 CALL VCHAR(L,M(L),40,3):: PR=M(L):: D=-D :: NEXT L :: RETURN
940 CALL DELSPRITE(#1):: FOR L=6 TO 18 STEP 3
950 CALL VCHAR(L,M(L),INT(RND*(143-140+1))+140):: CALL VCHAR(L+1,M(L),INT(RND*(1
39-136+1))+136):: CALL VCHAR(L+2,M(L),32):: NEXT L :: CALL FF :: RETURN
960 PU=INT(RND*(100-20+1))+20 :: IF PU<50 THEN 980 ELSE CALL SOUND(250,261,9,329
,9,391,9):: CALL SOUND(250,349,12,440,12,523,12)
970 CALL SOUND(250,391,14,493,14,587,14):: CALL SOUND(4100,1046,16,783,16,329,16
)
980 ON INT(RND*17)+1 GOTO 990,1000,1010,1020,1030,1040,1050,1060,1070,1080,1090,
1100,1110,1120,1130,1140,1150
990 B$="0000206028381030" :: GOTO 1160
1000 B$="00000000D8F87020" :: GOTO 1160
1010 B$="0000405864687060" :: GOTO 1160
1020 B$="000000000244F828" :: GOTO 1160
1030 B$="0000001028542810" :: GOTO 1160
1040 B$="0000101038282838" :: GOTO 1160
1050 B$="0A0C0828107C3838" :: GOTO 1160
1060 B$="0000000088708800" :: GOTO 1160
1070 B$="0000407F0F060600" :: GOTO 1160
1080 B$="0000183C7F000000" :: GOTO 1160
1090 B$="0000103892643810" :: GOTO 1160
1100 B$="000000107C443810" :: GOTO 1160
1110 B$="006000607F3C24AC" :: GOTO 1160
1120 B$="0000C6653C18081C" :: GOTO 1160
1130 B$="0000002828543810" :: GOTO 1160
1140 B$="00090600384C5438" :: GOTO 1160
```



```
1150 B$="00000000C8A7C486C"
1160 CALL CHAR(88,B$):: VS,ST=0 :: FOR SYMB=5 TO 17 STEP 3 :: FOR PLATZ=1 TO 3
1170 VS=INT(RND*(30-8+1))+8 :: CALL GCHAR(SYMB,VS,ST)
1180 IF ST=32 THEN CALL HCHAR(SYMB,VS,88)ELSE 1170
1190 NEXT PLATZ :: NEXT SYMB :: RETURN
1200 S6=4 :: READ S1,S2,S3,S4,S5,S7 :: IF S7=5 THEN RESTORE 1690
1210 S1=S1+LEV :: S2=S2+LEV :: S3=S3+LEV :: S4=S4+LEV :: S5=S5+LEV :: S6=S6+LEV
1220 CALL MOTION(#28,0,-S6,#2,0,-S1,#3,0,S2,#4,0,-S3,#5,0,S4,#6,0,-S5):: GOTO 75
0
1230 CALL MOTION(#28,0,S6,#2,0,S1,#3,0,-S2,#4,0,S3,#5,0,-S4,#6,0,S5):: GOTO 750
1240 CALL DELSPRITE(#2,#3,#4,#5,#6,#28):: DISPLAY AT(20,1)SIZE(5):" "
1250 CALL HCHAR(20,4,43,3):: FOR I=1 TO 3 :: FOR A=0 TO 30 STEP 5 :: CALL SOUND(
-99,698,A,1924,A)
1260 NEXT A :: CALL CHAR(43,K2$):: FOR A=0 TO 30 STEP 5 :: CALL SOUND(-99,554,A,
1527,A):: NEXT A :: CALL CHAR(43,K1$):: NEXT I :: CALL HCHAR(20,4,32,3)
1270 FOR I=2 TO 6 :: CALL MOTION(#I,0,0):: NEXT I :: CALL SOUND(-4250,-4,0):: FO
R LI=EX TO 16 STEP -2 :: CALL LOCATE(#7,8,LI):: NEXT LI
1280 CALL SOUND(-4250,-3,9):: FOR RU=8 TO 152 STEP 2 :: CALL LOCATE(#7,RU,16)::
NEXT RU
1290 CALL SOUND(-4250,-1,9):: FOR RE=16 TO 48 :: CALL LOCATE(#7,152,RE):: NEXT R
E
1300 CALL SOUND(-4250,-3,6):: FOR LIZ=48 TO 16 STEP -1 :: CALL LOCATE(#1,152,LIZ
,#7,152,LIZ):: NEXT LIZ
1310 CALL SOUND(-4250,-2,8):: FOR HOZ=152 TO 8 STEP -2 :: CALL LOCATE(#1,HOZ,16,
#7,HOZ,16):: NEXT HOZ
1320 CALL SOUND(-4250,-4,1):: FOR REZ=16 TO EX STEP 2 :: CALL LOCATE(#1,8,REZ,#7
,8,REZ):: NEXT REZ :: CALL DELSPRITE(#1):: CALL SOUND(-1,40000,30)
1330 SC=SC+NU :: DISPLAY AT(23,22)SIZE(5):SC :: NU=0 :: DISPLAY AT(22,10)SIZE(5)
:NU :: CALL HCHAR(2,INT(EX/8)+1,105):: EX=EX-16 :: CALL LOCATE(#7,8,EX)
1340 T=300 :: CALL SOUND(T,262,2):: CALL SOUND(T,294,2):: CALL SOUND(2*T,330,2):
: CALL SOUND(3*T/4,349,2,440,2):: CALL SOUND(T/4,392,2)
1350 CALL SOUND(T/2,440,2):: CALL SOUND(T/2,494,2,391,2):: CALL SOUND(T,523,5,39
1,4,329,0)
1360 LEV=LEV+1 :: RN=RN+1 :: IF RN=10 THEN 1380
1370 CALL LEER :: DISPLAY AT(20,1)SIZE(5):"LIFT+" :: H=32 :: V=40 :: GOSUB 940 :
: GOSUB 890 :: GOSUB 960 :: GOTO 510
1380 GOSUB 940 :: DISPLAY AT(5,6)SIZE(21):"SIE HABEN ALLE SAECKE" :: DISPLAY AT(
8,6)SIZE(23):"GUT NACH OBEN GEBRACHTV"
1390 DISPLAY AT(11,6)SIZE(19):"ES KANN NOCH BESSER" :: DISPLAY AT(14,6)SIZE(20):
"WERDENV     ALSO LOSV"
1400 DISPLAY AT(17,6)SIZE(23):"PROBIEREN SIE ES BITTEV"
1410 CALL SOUND(700,4000,30):: T=450 :: CALL SOUND(T,330,2,131,6):: CALL SOUND(T
,330,3,131,6):: CALL SOUND(T,349,3,131,6)
1420 CALL SOUND(T,392,2,131,6):: CALL SOUND(T,392,3,147,6):: CALL SOUND(T,349,3,
147,6)
1430 CALL SOUND(T,330,3,147,6):: CALL SOUND(T,294,3,147,6):: CALL SOUND(T,262,2,
165,6)
1440 CALL SOUND(T,262,3,165,7):: CALL SOUND(T,294,2,175,6):: CALL SOUND(T,330,2,
176,5)
1450 CALL SOUND(T*1.5,294,2,196,7):: CALL SOUND(T/2,262,3,196,7):: CALL SOUND(4*
T,262,4,165,7,131,8):: CALL FF :: CALL FF
1460 IF SC<=NSC THEN 1620
1470 NSC=SC :: CALL SOUND(800,4444,30):: DISPLAY AT(23,10)SIZE(5):NSC
1480 T=500 :: U=T*3/4 :: CALL DELSPRITE(#7)
1490 CALL LEER :: CALL SOUND(1,9999,30):: CALL SOUND(U,156,6):: DISPLAY AT(5,6)S
IZE(24):"DAS IST EINE NEUE" :: CALL SOUND(T/4,208,5)
1500 CALL SOUND(T,262,3):: CALL SOUND(U,262,4):: DISPLAY AT(8,6)SIZE(24):"BESTLE
ISTUNGV DAFUER" :: CALL SOUND(T/4,233,4)
1510 CALL SOUND(U,208,3):: DISPLAY AT(11,6)SIZE(20):"EINE SONDERMELODIEV" :: CAL
L SOUND(T/4,262,4):: CALL SOUND(T+U,175,3,139,10)
1520 DISPLAY AT(14,6)SIZE(24):"MACHEN SIE WEITER SOV" :: CALL SOUND(T/4,220,3)::
 CALL SOUND(U,233,3):: CALL SOUND(T/4,262,2):: CALL SOUND(U,277,2,233,8)
1530 DISPLAY AT(17,6)SIZE(24):"STEIGERN SIE SICH" :: CALL SOUND(T/4,262,3):: CAL
```

```
! SOUND(U,233,2,196,8):: CALL LEER :: CALL SOUND(T/4,277,2)
1540 DISPLAY AT(5,6)SIZE(24):"ES GEHT GLEICH WEITER" :: CALL SOUND(U,196,2,156,8
):: CALL SOUND(T/4,233,2):: CALL SOUND(U,165,3,131,8)
1550 DISPLAY AT(8,6)SIZE(24):"MIT EINER NEUEN RUNDEV"
1560 CALL SOUND(T/4,196,3):: CALL SOUND(T+U,262,2):: CALL SOUND(T/4,233,4):: CAL
L SOUND(U,220,4,175,10)
1570 DISPLAY AT(11,6)SIZE(24):"ABER BITTE NUR DANN"
1580 CALL SOUND(T/4,175,4):: CALL SOUND(U,233,2,117,10):: CALL SOUND(T/4,277,3):
: DISPLAY AT(14,6)SIZE(24):"WENN SIE DAS WOLLENV"
1590 CALL SOUND(U,196,4,156,10):: CALL SOUND(T/4,156,4):: CALL SOUND(2*T,208,2):
: DISPLAY AT(17,6)SIZE(24):"ES LIEGT NUR AN IHNENV" :: GOTO 1630
1600 GOSUB 940 :: DISPLAY AT(5,7)SIZE(19):"LEIDER HAT ES NICHT" :: DISPLAY AT(8,
7)SIZE(9):"GEKLAPPTV" :: DISPLAY AT(11,7)SIZE(20):"LASSEN SIE BITTE DEN"
1610 DISPLAY AT(14,7)SIZE(22):"KOPF´NICHT HAENGEN UND" :: DISPLAY AT(17,7)SIZE(2
1):"WAGEN SIE ES NOCHMALV"
1620 DISPLAY AT(23,10)SIZE(5):NSC
1630 FOR I=2 TO 5 :: CALL SPRITE(#I,96,FA,152,256,0,INT(RND*35+8)):: NEXT I
1640 DISPLAY AT(20,7)SIZE(20):"NEUES SPIEL J / N ?" :: CALL FF
1650 CALL KEY(3,T,S):: IF T=78 THEN END ELSE IF T=74 THEN 1660 ELSE DISPLAY AT(2
0,7)SIZE(20):RPT$(CHR$(88)&CHR$(32),10):: CALL FF :: GOTO 1640
1660 LEV,NU,TOT,SC,RN=0 :: EX=232 :: CALL LEER :: DISPLAY AT(22,10)SIZE(5):NU ::
 DISPLAY AT(23,23)SIZE(5):SC
1670 CALL HCHAR(2,5,32,26):: CALL HCHAR(22,25,113,5):: GOSUB 890 :: GOSUB 1160 :
: DISPLAY AT(20,1)SIZE(5):"LIFT+" :: GOTO 500
1680 DATA 2,4,4,2,3,0,2,2,4,4,3,0,3,4,3,2,2,0,2,2,3,3,4,0,3,3,4,3,4,0,2,3,3,3,4,
0
1690 DATA 2,4,3,3,4,0,2,3,3,4,2,0,2,4,2,4,2,0,3,2,1,4,3,0,2,4,2,3,4,0,2,4,3,2,4,
5
1700 SUB FF :: FB=INT(RND*(14-3+1))+3
1710 FC=INT(RND*(14-3+1))+3 :: IF FC=FB THEN 1710
1720 FD=INT(RND*(14-3+1))+3 :: IF FD=FC OR FD=FB THEN 1720
1730 FE=INT(RND*(14-3+1))+3 :: IF FE=FD OR FE=FC OR FE=FB THEN 1730
1740 FG=INT(RND*(14-3+1))+3 :: IF FG=FC OR FG=FB OR FG=FE OR FG=FD THEN 1740
1750 CALL COLOR(12,FC,2,13,FC,2,14,FB,2,2,FE,2,10,FD,2,8,FG,2):: SUBEND
1760 SUB TASTER
1770 DISPLAY AT(24,2):"JOYSTICK-TASTER 2 DRUECKEN"
1780 CALL KEY(2,T,S):: IF S=0 OR T<>18 THEN 1770
1790 CALL CLEAR :: SUBEND
1800 SUB LEER :: FOR I=5 TO 20 STEP 3 :: CALL HCHAR(I,8,32,23):: NEXT I :: SUBEN
D
```





Computercamp
Ferienzentrum Schloß Dankern

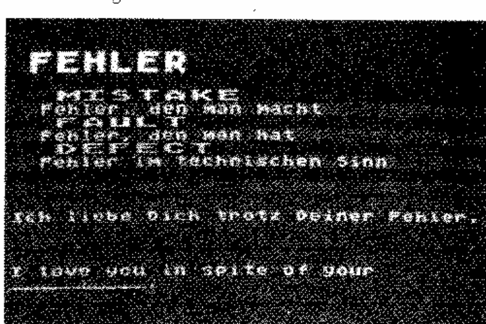
*Einsteiger, Fortgeschrittene und „Cracks" werden bei uns von qualifizierten Pädagogen bzw. Informatikern betreut, die es verstehen, **individuell** auf den Wissensstand jedes Teilnehmers einzugehen und Informationen **spielerisch** zu vermitteln.*



Die angebotene Palette umfaßt:

- *Einführung in Hardware und -Bedienung*
- *Einführung in die Kommunikation mit dem Computer*
- *Einführung in die Programmiersprachen*
- *Die Programmiersprache BASIC von A–Z*
- *BASIC für Fortgeschrittene*
- *Perfektionieren von Programmen in BASIC*
- *Einführung und Programmieren in Maschinensprache*

*Darüberhinaus aber natürlich **jede Menge detaillierter Information** wie zum Beispiel BASIC-Dialekte, oder wie man einen bereits vorhandenen Computer optimal nutzen kann, etc.*
*Da der Erfolg der Kurse auch von der Verfügbarkeit der Geräte abhängt, **garantieren wir jedem Teilnehmer ein eigenes Gerät,** das er mindestens **3 Stunden täglich** nutzen kann.*
Und – wie gesagt – für individuelle Betreuung ist genauso gesorgt wie für Raum zur Entfaltung von Kreativität und Eigeninitiative – das fördert den wichtigen Erfahrungsaustausch.
*Zum Abschluß des Camps erhält jeder Teilnehmer ein **Abschlußzertifikat,** und kann natürlich auch selbst erstellte Programme etc. mit nach Hause nehmen.*



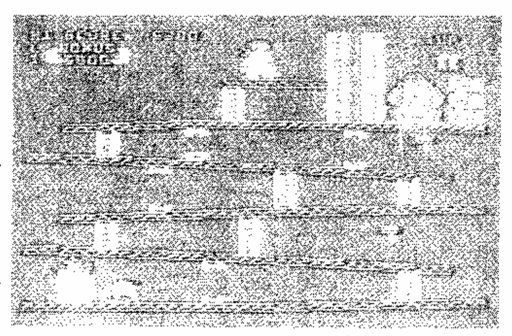
Wir freuen uns schon auf Sie – bis bald!

*Fast zuviel, um alles aufzuzählen: **Ponyreiten, Schwimmbad, Minigolf, Tischtennis, Riesenspielplatz** mit **Tarzanschaukel, Westernfort, Fahrradkarussel, Pferderennen, Kletterturm, Riesenrutsche... alles im Preis enthalten!** Darüberhinaus **Kegelbahnen, Autoscooter, Modellflugplatz, Fernsehräume,** und auf dem Wasser **Wasserskilift, Windsurfen, Wasserfahrräder, Tret- und Ruderboote...***

*Nicht zu vergessen das 300 Jahre alte **Schloß,** das zur Besichtigung einlädt, genauso wie mehrere **Cafés, Restaurants,** ein **Ferienkino,** eine **Super-Disco...***

Haben wir etwas vergessen? Ach ja – einkaufen kann man auch – sogar sonntags!

Wir finden: Ein insgesamt überzeugendes Angebot, das sicher auch Computerfans begeistern wird.



Die Camps beginnen jeweils an einem Samstag. Die Belegung ist möglich für 1 Woche, 14 Tage oder 3 Wochen (Pfingsten nur 4 Tage).

Ostern '84
14.4.–21.4.
21.4.–28.4.

Pfingsten '84
9.6.–12.6.

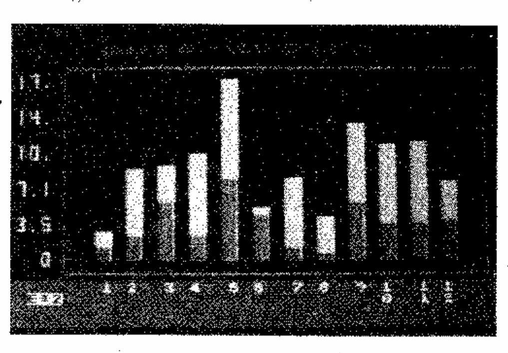
Sommer '84
23.6.–30.6.
30.6.– 7.7.
7.7.–14.7.
14.7.–21.7.
21.7.–28.7.
28.7.– 4.8.
4.8.–11.8.
11.8.–18.8.
18.8.–25.8.
25.8.– 1.9.
1.9.– 8.9.

Herbst '84
6.10.–13.10.
13.10.–20.10.
20.10.–27.10.
27.10.– 3.11.



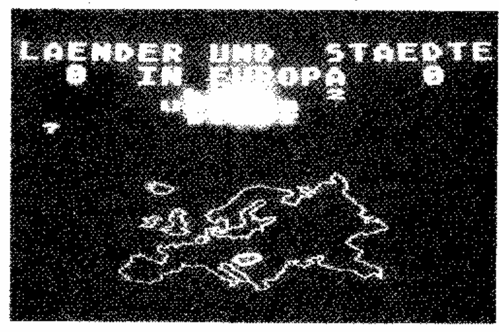
4 Tage (nur Pfingsten)	350,– DM
1 Woche	550,– DM
2 Wochen	980,– DM
3 Wochen	1480,– DM

*Die Preise gelten für Unterbringung **mit Vollpension; die Kurse und fast alle Freizeitmöglichkeiten sind inklusive.***
Sondertarife für Gruppen oder Schulklassen auf Anfrage. Eltern, die ihre Kinder begleiten wollen, sind dazu herzlich eingeladen – rufen Sie uns einfach an!



*Gewohnt wird nicht in Hotels, Jugendherbergen oder Zeltlagern, sondern in **supergemütlichen Ferienhäusern** (4–6 Personen) aus Holz, die in unmittelbarer Nähe des Schlosses Dankern und des dazugehörigen Dankern-Sees liegen, inmitten eines **150 Hektar großen Ferienzentrums.** Übrigens können Eltern auch gerne ihre Kinder begleiten – Anfrage beim Buchungsbüro genügt.*

Frühstück, Mittag- und Abendessen sind inklusive.





*Ort der Handlung ist das **Ferienzentrum Schloß Dankern,** 4472 Haren (Ems). Ein riesiges, landschaftlich wunderschön gelegenes Feriengebiet mitten im schönen Emsland. Wie man hinkommt, ist am besten der Karte zu entnehmen. Die Bahn (Bahnhof Haren) ist nur etwa 2 km entfernt. Weitere Fragen beantworten wir gerne – rufen Sie uns bitte an.*

Buchungsbüro Computercamp
Ferienzentrum Schloß Dankern
Holztwiete 4 D
2000 Hamburg 52
Tel.: (040) 82 79 42

Dragon 32-64

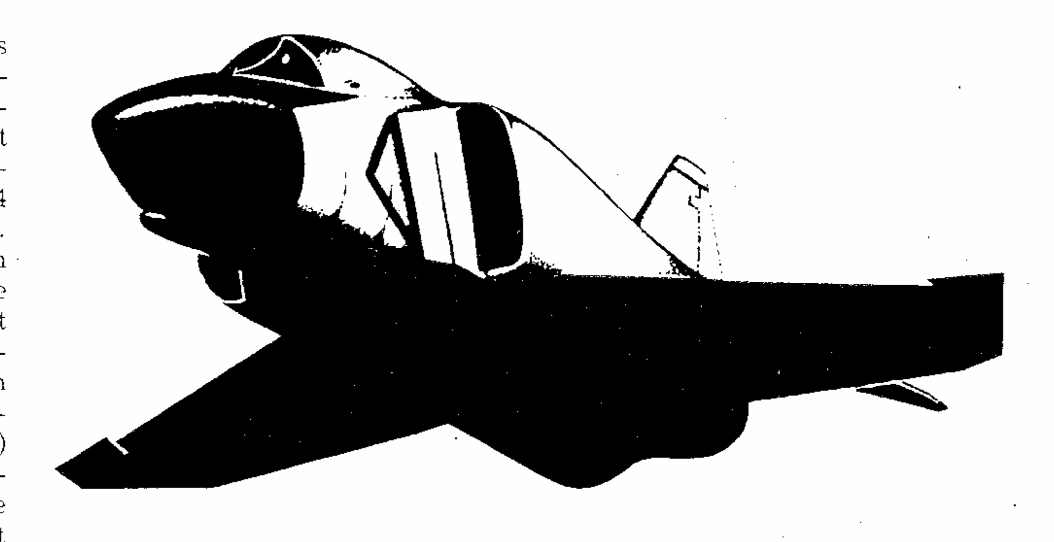
Wargames

WARGAMES

Ein spannendes Simulationsspiel

Wie der Name schon sagt, handelt es sich hierbei um ein recht kriegerisches Spiel, bei dem taktisches Können und Glück beim ‚Würfeln' gefragt sind. Es geht darum, die aus 16 Ländern bestehende (fiktive) Welt, die in 4 Kontinente aufgeteilt ist, zu erobern. Nach Eingabe des Programms kann das Spiel beginnen (am besten sollte das Programm nach der doch recht mühseligen Tipparbeit zunächst abgespeichert werden), und zwar durch Eingabe der BASIC- (Baginner's All-Purpose Symbolic Instruction Code) Vokabel „RUN". Der Computer erstellt zunächst ein Titelbild (sofern Sie sich bei der Eingabe des Listings nicht vertippt haben) und meldet sich anschließend mit der Frage nach der Anzahl der Mitspieler (2–4). Übrigens: Sobald Sie in irgendeiner Situation eine Fehleingabe machen, zeigt der Computer dies durch einen Signalton und durch Wiederholung der Fragestellung an. Anschließend muß jeder seinen Namen und ein für seine Truppen charakteristisches Zeichen angeben. Nach Erledigung dieser Formalitäten erscheint das eigentliche Spielfeld, nachdem die Länder gerecht verteilt wurden. Jeder Mitspieler kann unschwer die ihm gehörenden Länder ausmachen, denn diese sind durch das Truppenkennzeichen deutlich gemacht. Die Länder enthalten jeweils drei zur Identifikation nötige Elemente, nämlich (1.) die Nummer des Landes, (2.) das Truppenkennzeichen des Besitzers und (3.) die Zahl der dort stationierten Truppen.

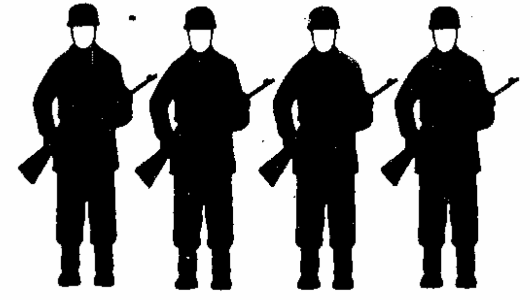
Die erste Frage, die dem Spieler der gerade an der Reihe ist, gestellt wird, ist die, wo seine Truppen positioniert werden sollen. Die Anzahl der zugeteilten Truppen hängt von der Zahl der eroberten Kontinente, der der eroberten Länder sowie einem gerechten Zufallsprinzip ab. Man kann die so zur Verfügung gestellten Truppen auf ein oder mehrere Länder verteilen, gerade so, wie es zum Angriff oder zur Verteidigung am günstigsten erscheint. Allerdings darf die Zahl der Truppen im Zielland 99 nicht überschreiten. Sobald alle Truppen aufgeteilt sind, gelangt man in die Angriffsphase.



Man kann nun entweder durch Eingabe von ‚EN' die Angriffsphase verlassen und sich auf friedlichere Dinge konzentrieren oder aber die Länder, die unmittelbar an eigene grenzen, angreifen und zu erobern hoffen. Dies geschieht durch Eingabe der Nummer des angreifenden Landes. Natürlich darf man von einem Land aus nur dann angreifen, wenn mehr als eine Truppe dort stationiert ist, sonst müßte man das angreifende Land aufgeben, um das neue zu besetzen. Sofern diese Bedingung jedoch erfüllt ist, erscheint das Truppenkennzeichen des Angreifers, um darauf aufmerksam zu machen, daß dieser mit dem ‚Würfeln' an der Reihe ist.

Dies geschieht durch Eingabe des Truppenkennzeichens. Man kann jeweils nur mit höchstens drei Armeen gleichzeitig angreifen; das gleiche gilt für die Verteidigung. Nachdem der Angreifer fertig gewürfelt hat, ist der Verteidiger an der Reihe (man kann übrigens auch mit einer Truppe verteidigen!). Für jeden Truppenverlust seitens des Verteidigers ertönt nach dem Würfeln ein tiefer, für jeden des Angreifers ein hoher Ton. Sofern der Angreifer das Land einnehmen kann, ertönt eine Tonfolge. Die Verluste werden dadurch errechnet, daß zunächst die gewürfelten Zahlen der Größe nach sortiert und anschließend miteinander verglichen werden, das heißt: der höchste Wurf des Angreifers wird mit dem höchsten des Verteidigers, der zweithöchste des Angreifers mit dem zweithöchsten des Verteidigers verglichen und so weiter. Bei jedem so gewonnenen Vergleich wird dem Angreifer eine Truppe abgezogen, falls seine Zahl kleiner oder gleich der des Verteidigers ist, anderenfalls verliert der Verteidiger eine Truppe. Sobald letzterer keine Truppen mehr in seinem Land hat, gilt dieses als erobert. Das heißt, der Angreifer darf dort mit einer von ihm festgelegten Truppenzahl einmarschieren. Nach einem solchen Kampf gelangt man wieder zum Anfang der Angriffsphase. Will man nochmals vom selben Land aus dasselbe feindliche angreifen, kann man nun ‚WH' eingeben, anderenfalls verfährt man wie bereits am Anfang dieses Abschnitts beschrieben.

Sobald man mit den Angriffen fertig ist, kann man noch abschließend seine Truppen aus einem Land in ein anderes verschieben, so wie es strategisch am günstigsten erscheint. Dieser Programmteil bedarf keiner näheren Erläuterung, man muß lediglich



darauf achten, daß (1.) beide Länder das Eigentum des Spielers sind, (2.) daß im Ausgangsland der Verschiebung mindestens eine Truppe verbleibt und (3.) daß die Zahl der Truppen im Zielland 99 nicht übersteigt. Auch diesen Modus verläßt man durch Eingabe von ‚EN'.

Hiernach ist der nächste Spieler an der Reihe. Sobald ein Spieler keine Länder mehr besitzt, wird dieser aus der Spielreihenfolge gestrichen; erreicht jemand die Länderzahl von 16, was der Weltherrschaft entspricht, wird er entsprechend geehrt.

Taktische Hinweise:

Man sollte sich darum bemühen, jede Runde mindestens ein Land zu erobern, denn dadurch erhöht sich die Wahrscheinlichkeit für einen ‚Truppenbonus' nach dem bereits anfangs erwähnten Zufallsprinzip. Man sollte darauf achten, seine Kontinente (sofern vorhanden) stets ausreichend nach außen hin abzusichern und notfalls auf die Eroberung eines Landes verzichten, um die Verteidigung wichtiger Gebiete zu garantieren. Die Kontinente haben unterschiedliche, von der Schwierigkeit beim Einnehmen und Halten abhängige Wertungen. Für die Haltung des ersten Kontinents (Länder 1–4) gibt es pro Runde 7, für den zweiten (5–8) ebenso wie für den vierten (13–15) 3 und für den dritten 5 zusätzliche Armeen pro Runde. Viel Spaß beim Spiel!

WARGAMES: 8.4KBYTES



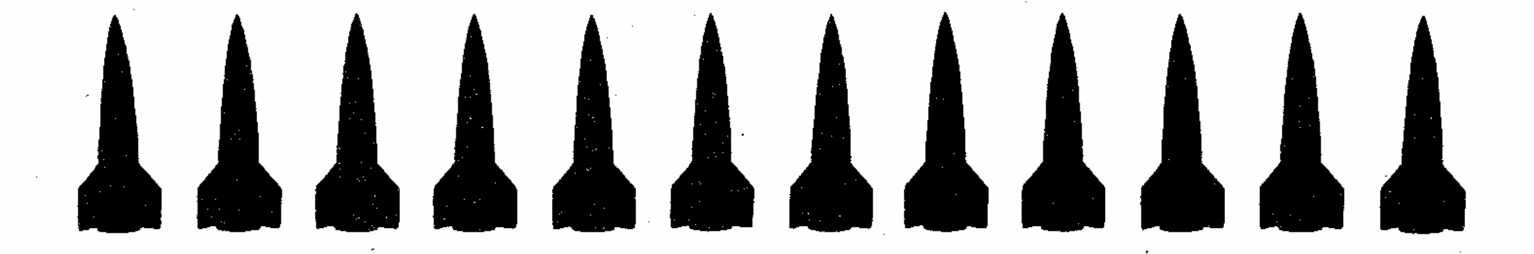
```
0 REM ***************************
1 REM **        WARGAMES       **
2 REM ***************************
3 REM **  COPYRIGHT (C) 1984   **
4 REM **          BY           **
5 REM **  JOERG       TEGEDER  **
6 REM **  KEPLERSTRASSE     5  **
7 REM **  5206    NEUNKIRCHEN  **
8 REM **  TELEFON:02247/4147   **
9 REM **         UND           **
10 REM**  ANDREAS    HALLWACHS **
11 REM**  KOENIGSBERGERSTR.4   **
12 REM**  5206    NEUNKIRCHEN  **
13 REM**  TELEFON:02247/5178   **
14 REM***************************
50 '
100 OP=0:SC=8:AR=0:GOSUB 11700
200 CLS 8:TE$="WIEVIELE SPIELER (2-4) ?":PO=67:LE=1:GOSUB 15900:SP=VAL(A$):IF SP
<2 OR SP>4 THEN SOUND 150,2:GOTO 200
300 CLS 4:PRINT@260,"IHR NAME,";:FOR A=1 TO SP
400 TE$="SPIELER"+STR$(A)+"?":LE=8:PO=328:GOSUB 15900:NA$(A)=A$:IF NA$(A)="" THE
N SOUND 150,2:GOTO 400 ELSE IF A=1 THEN NEXT A
500 IF A>1 THEN FOR QW=1 TO A-1:IF NA$(A)=NA$(QW) THEN QW=A:NEXT QW:SOUND 150,2:
GOTO 400 ELSE NEXT QW,A
600 FOR A=1 TO SP:CLS 3:PRINT@100,"GEBEN SIE IHR TRUPPEN-";:PRINT@140,"KENNZEICH
EN EIN!";
700 TE$=NA$(A)+"? ":LE=1:PO=200:GOSUB 15900:TR$(A)=A$:IF TR$(A)="" THEN SOUND 15
0,2:GOTO 700 ELSE IF ASC(TR$(A))<58 AND ASC(TR$(A))>47 THEN SOUND 150,2:GOTO 700
800 IF A>1 THEN FOR B=1 TO A-1:IF TR$(B)=TR$(A) THEN B=A:NEXT B:SOUND 150,2:GOTO
 700 ELSE NEXT B
900 NEXT A
1000 GOSUB 6000
1100 FOR A=1 TO SP:LA(A)=0:NEXT A
1200 FOR LV=1 TO 16:LA(0)=17:MR=0:FOR A=SP TO 1 STEP -1:IF LA(A)<LA(MR) THEN MR=
A
1300 NEXT A:LA(MR)=LA(MR)+1:DA(LV,1)=MR:DA(LV,2)=1:NEXT LV
1400 FOR C=1 TO 16:GOSUB 17400:NEXT C
1500 FOR B=1 TO SP:IF LA(B)=0 THEN NEXT B:GOTO 1500 ELSE CH=0
1600 TR=0:IF ZU(B)>5 THEN TR=GR+RND(3):GR=GR+2
1700 FOR A=1 TO 4:ZW=0:FOR W=(A-1)*4+1 TO (A-1)*4+4:IF DA(W,1)=B THEN ZW=ZW+1
1800 NEXT W:IF ZW=4 THEN IF A=1 THEN TR=TR+7 ELSE IF A=2 THEN TR=TR+3 ELSE IF A=
3 THEN TR=TR+5 ELSE TR=TR+3
```

```
1900 NEXT A
2000 GOSUB 16900:TR=TR+INT(LA(B)/6)+3:A=TR:GOSUB 17300:PRINT@384,NA$(B)" ("TR$(B
)"), SIE HABEN "Z$;:PRINT@416,"TRUPPEN EINZIEHEN KOENNEN.";
2100 PRINT@448,"TRUPPEN=";:A=TR:GOSUB 17300:PRINTZ$;:TE$="LAND NR. ==>":PO=480:L
E=2:GOSUB 15900:PRINT@480,STRING$(31,32);:GOSUB 16600
2200 C=VAL(A$):IF C>16 OR C<1 THEN SOUND 150,2:GOTO 2100 ELSE IF DA(C,1)<>B THEN
 SOUND 150,2:GOTO 2100
2300 TE$="ZAHL DER TRUPPEN ==>":LE=2:PO=480:GOSUB 15900:PRINT@480,STRING$(31,32)
;:GOSUB 16600:D=VAL(A$):IF D<1 OR D>TR OR DA(C,2)+D>99 THEN SOUND 150,2:GOTO 230
0
2400 TR=TR-D:DA(C,2)=DA(C,2)+D:GOSUB 17400:IF TR>0 THEN GOTO 2100
2500 C=0:AN=0
2600 GOSUB 16900:PRINT@384,"'EN'=ENDE,''WH'=WIEDERHOLUNG";:TE$="ANGRIFF VON LAND
 NR. ==>":PO=416:LE=2:GOSUB 15900:PRINT@416,STRING$(32,32);:GOSUB 16600:IF A$="E
N" THEN GOTO 4200 ELSE IF A$="WH" THEN GOTO 5600 ELSE C=VAL(A$)
2700 IF C<1 OR C>16 THEN SOUND 150,2:GOTO 2600
2800 IF DA(C,1)<>B OR DA(C,2)<2 THEN SOUND 150,2:GOTO 2600 ELSE AN=C
2900 GOSUB 16900:TE$="NACH LAND NR. ==>":PO=416:LE=2:GOSUB 15900:PRINT@416,STRIN
G$(32,32);:GOSUB 16600:C=VAL(A$):IF C<1 OR C>16 THEN SOUND 150,2:GOTO 2900 ELSE
IF DA(C,1)=B THEN SOUND 150,2:GOTO 2900 ELSE Z=3
3000 IF C=DA(AN,Z) THEN GOTO 3100 ELSE Z=Z+1:IF DA(AN,Z)=0 THEN SOUND 150,2:GOTO
 2900 ELSE GOTO 3000
3100 GOSUB 16900:PRINT@384,"ANGRIFF: "NA$(DA(AN,1))" ==> "NA$(DA(C,1));:GOSUB 16
600
3200 IF DA(AN,2)<4 THEN K=DA(AN,2)-1 ELSE K=3
3300 TE$="ANGREIFER:":PO=416:ZW=AN:GOSUB 5400
3400 AK=K:FOR A=1 TO K:W1(A)=W(A):NEXT A
3500 IF DA(C,2)<3 THEN K=DA(C,2) ELSE K=3
3600 SOUND 200,5:TE$="VERTEIDIGER:":PO=448:ZW=C:GOSUB 5400
3700 IF AK>K THEN PR=K ELSE PR=AK
3800 FOR A=1 TO PR:IF W1(A)>W(A) THEN DA(C,2)=DA(C,2)-1:SOUND 25,5 ELSE DA(AN,2)
=DA(AN,2)-1:SOUND 225,5
3900 NEXT A:IF DA(C,2)=0 THEN PLAY P$:CH=1:LA(DA(AN,1))=LA(DA(AN,1))+1:LA(DA(C,1
))=LA(DA(C,1))-1
4000 IF DA(C,2)=0 THEN DA(C,1)=DA(AN,1):GOSUB 17400:TE$="TRUPPEN NACH LAND NR. "
:A=C:GOSUB 17300:TE$=TE$+Z$+" ==>":PO=480:LE=2:GOSUB 15900:A=VAL(A$):IF A<1 OR A
=>DA(AN,2) THEN SOUND 150,2:GOTO 4000 ELSE DA(C,2)=A:DA(AN,2)=DA(AN,2)-A
4100 GOSUB 17400:ZW=C:C=AN:GOSUB 17400:C=ZW:GOTO 2600
4200 GOSUB 16900:IF LA(B)=16 THEN GOTO 5000 ELSE PRINT@384,"TRUPPENVERSCHIEBUNGE
N---EN=ENDE";:TE$="VON LAND NR. ==>":LE=2:PO=448:GOSUB 15900:IF A$="EN" THEN GOT
O 4700 ELSE A=VAL(A$):IF A<1 OR A>16 OR DA(A,1)<>B OR DA(A,2)<2 THEN SOUND 150,2
:GOTO 4200
4300 PRINT@448,STRING$(31,32);:GOSUB 16900:TE$="TRUPPEN NACH LAND NR. ==>":PO=41
6:LE=2:GOSUB 15900:DE=VAL(A$):IF DE<1 OR DE>16 OR DA(DE,1)<>B OR DA(DE,2)+ZT>99
THEN SOUND 150,2:GOTO 4200 ELSE Z=3
4400 IF DE=DA(A,Z) THEN GOTO 4500 ELSE Z=Z+1:IF DA(A,Z)=0 THEN SOUND 150,2:GOTO
4200 ELSE GOTO 4400
4500 GOSUB 16900:TE$="ZAHL DER TRUPPEN ==>":PO=480:LE=2:GOSUB 15900:ZT=VAL(A$):I
F ZT<1 OR DA(A,2)<ZT+1 THEN SOUND 150,2:GOTO 4200
4600 DA(DE,2)=DA(DE,2)+ZT:DA(A,2)=DA(A,2)-ZT:C=A:GOSUB 17400:C=DE:GOSUB 17400:GO
TO 4200
4700 IF CH=1 THEN ZU(B)=ZU(B)+RND(3):CH=0
4800 NEXT B
4900 GOTO 1500
5000 SC=3:GOSUB 11700:PRINT@389,"WELTBEHERRSCHER IST:";
5100 FOR C=1 TO 3:PRINT@448+(32-LEN(NA$(B)))/2,NA$(B);:FOR A=1 TO 200 STEP 10:SO
UND A,1:NEXT A:PRINT@448,STRING$(32,128);:FOR A=200 TO 1 STEP -10:SOUND A,1:NEXT
 A,C
5200 PRINT@416+(32-LEN(NA$(B)))/2,NA$(B);:FOR A=1 TO 200:NEXT A:TE$="NOCHMAL J/N
 ?":PO=488:LE=1:GOSUB 15900:IF A$="J" THEN RUN ELSE IF A$<>"N" THEN GOTO 5200
5300 CLS 0:FOR A=10 TO 1 STEP -1:GOSUB 17300:PRINT@239,Z$;:SOUND (210-A*20),1:FO
```

```
R B=1 TO 100:NEXT B,A:CLS:PRINT@134,"TSCHUESS...":PRINT@384,;:END
5400 FOR WF=1 TO K:GOSUB 5700:W(WF)=A:IF WF>1 AND W(WF-1)<A THEN FOR Z=1 TO WF:I
F W(Z)>A THEN NEXT Z ELSE X=Z:Z=WF:NEXT Z:FOR Z=WF TO X+1 STEP -1:W(Z)=W(Z-1):NE
XT Z:W(X)=A
5500 FOR Z=1 TO WF:PRINT@PO,TE$;:A=W(Z):GOSUB 17300:PRINT@PO+10+Z*4,Z$;:GOSUB 16
600:NEXT Z,WF:RETURN
5600 IF AN<1 OR AN>16 OR DA(AN,1)<>B OR DA(AN,2)<2 THEN SOUND 150,2:GOTO 2600 EL
SE IF DA(C,1)=B THEN SOUND 150,2:GOTO 2900 ELSE GOTO 3100
5700 A$=INKEY$:A=RND(0):PRINT@PO+10+WF*4,TR$(DA(ZW,1));:GOSUB 16600
5800 A$=INKEY$:IF A$<>TR$(DA(ZW,1)) THEN GOTO 5800
5900 A=RND(6):GOSUB 17300:RETURN
6000 CLS 0:FOR LI=0 TO 63:SET(LI,0,5):SET(LI,1,5):SET(LI,22,5):SET(LI,23,5):NEXT
 LI:FOR LI=0 TO 23:SET(0,LI,5):SET(1,LI,5):SET(62,LI,5):SET(63,LI,5):NEXT LI
6100 SOUND 10,1
6200 FOR LI=2 TO 14:SET(LI,5,2):SET(12+LI,7,2):SET(25+LI,13,2):SET(37+LI,17,2):S
ET(25+LI,8,2):SET(37+LI,10,2)
6300 IF LI<8 THEN SET(25,LI,2)
6400 SOUND 20,1
6500 IF LI<8 THEN SET(13,LI,2)
6600 IF LI<5 THEN SET(40,18-LI,2)
6700 SOUND 30,1
6800 IF LI<8 THEN SET(51,18-LI,2)
6900 IF LI<14 THEN SET(26,LI,2)
7000 SOUND 40,1
7100 IF LI<6 THEN SET(39,14-LI,2)
7200 NEXT LI
7300 SOUND 50,1
7400 SET(39,9,2)
7500 SET(14,6,2)
7600 SOUND 60,1
7700 FOR LI=2 TO 13:SET(26+LI,7,8):SET(38+LI,9,8)
7800 IF LI<9 THEN SET(51,LI,8) ELSE SET(52,LI,8)
7900 SOUND 70,1
8000 IF LI<8 THEN SET(26,LI,8)
8100 IF LI<8 THEN SET(39,LI,8)
8200 SOUND 80,1
8300 IF LI<12 THEN SET(50+LI,13,8)
8400 IF LI<6 THEN SET(44,16+LI,8)
8500 SOUND 90,1
8600 IF LI<9 THEN SET(43+LI,18,8)
8700 IF LI<8 THEN SET(52,11+LI,8)
8800 SOUND 100,1
8900 NEXT LI
9000 SET(27,7,8)
9100 SOUND 110,1
9200 SET(40,7,8)
9300 SET(40,8,8)
9400 SOUND 120,1
9500 FOR LI=0 TO 12:SET(38-LI,18,7):SET(39-LI,14,7):SET(26-LI,18,7):SET(2+LI,18,
7)
9600 IF LI<4 THEN SET(39,14+LI,7)
9700 SOUND 130,1
9800 IF LI<5 THEN SET(39+LI,18,7)
9900 IF LI<4 THEN SET(43,18+LI,7)
10000 SOUND 140,1
10100 IF LI<8 THEN SET(26,21-LI,7)
10200 IF 17+LI<22 THEN SET(14,17+LI,7)
10300 SOUND 150,1
10400 NEXT LI
10500 SET(26,14,7)
```

```
10600 SOUND 160,1
10700 FOR LI=0 TO 11:SET(2+LI,17,1):SET(13,17-LI,1):SET(13+LI,13,1):SET(14+LI,17
,1):SET(2+LI,13,1):SET(2+LI,6,1):SET(13+LI,8,1)
10800 IF LI<10 THEN SET(25,8+LI,1)
10900 SOUND 170,1
11000 NEXT LI
11100 SET(25,13,1)
11200 SOUND 180,1
11300 SET(25,8,1)
11400 PRINT @ 11,"wargames";:GOSUB 16600
11500 SOUND 190,1
11600 RETURN
11700 CLS 0
11800 FOR SD=2 TO 10:SET(20,SD,SC):SET(26,SD,SC)
11900 IF SD<6 THEN SET(23,2+SD,SC)
12000 IF SD<3 THEN SET(21,7+SD,SC):SET(22,6+SD,SC):SET(24,6+SD,SC):SET(25,7+SD,S
C)
12100 NEXT SD
12200 FOR SD=5 TO 9:SET(29,SD,SC):SET(32,SD,SC)
12300 IF SD<7 THEN SET(25+SD,4,SC):SET(25+SD,10,SC)
12400 IF SD<6 THEN SET(28+SD,10,SC)
12500 NEXT SD
12600 FOR SD=4 TO 10:SET(36,SD,SC)
12700 IF SD<5 THEN SET(33+SD,5,SC):SET(36+SD,5,SC)
12800 IF SD<6 THEN SET(34+SD,4,SC)
12900 NEXT SD
13000 FOR SD=1 TO 7:SET(12,13+SD,SC)
13100 IF SD<6 THEN SET(12+SD,13,SC):SET(12+SD,21,SC)
13200 IF SD<3 THEN SET(18,13+SD,SC):SET(18,21-SD,SC)
13300 IF SD<4 THEN SET(19-SD,18,SC)
13400 NEXT SD
13500 FOR SD=1 TO 5:SET(21,15+SD,SC):SET(24,15+SD,SC)
13600 IF SD<3 THEN SET(21+SD,15,SC):SET(21+SD,21,SC)
13700 IF SD<2 THEN SET(24+SD,21,SC)
13800 NEXT SD
13900 FOR SD=0 TO 5:SET(28,16+SD,SC):SET(31,16+SD,SC):SET(34,16+SD,SC)
14000 IF SD<2 THEN SET(29+SD,15,SC):SET(32+SD,15,SC)
14100 NEXT SD
14200 FOR SD=0 TO 4:SET(37,16+SD,SC)
14300 IF SD<3 THEN SET(38+SD,15,SC):SET(38+SD,21,SC):SET(38+SD,18,SC)
14400 IF SD<2 THEN SET(41,16+SD,SC)
14500 NEXT SD
14600 FOR SD=0 TO 2:SET(45+SD,15,SC):SET(45+SD,18,SC):SET(45+SD,21,SC)
14700 IF SD<2 THEN SET(44,16+SD,SC):SET(48,19+SD,SC)
14800 IF SD<1 THEN SET(48,16+SD,SC):SET(44,20+SD,SC)
14900 NEXT SD
15000 IF AR<>0 THEN GOTO 15800 ELSE AR=AR+1
15100 PRINT @ 393,"COPYRIGHT (C)";:PRINT @ 430,"BY";:PRINT @ 454,"TEGEDER & HALL
WACHS";
15200 GOSUB 17500
15300 FOR C=1 TO 25
15400 SOUND C*10,1
15500 IF D=1 THEN SCREEN 0,0:D=0 ELSE SCREEN 0,1:D=1
15600 FOR B=1 TO 50:NEXT B,C
15700 SCREEN 0,1:FOR C=1 TO 500:NEXT C
15800 RETURN
15900 A$=""
16000 PRINT@PO,TE$A$CHR$(207)STRING$(LE,32);:SCREEN 0,1
16100 B$=INKEY$:IF B$="" THEN GOTO 16100
16200 IF B$=CHR$(13) THEN RETURN ELSE IF B$=CHR$(8) THEN IF PEEK(VARPTR(A$))>0 T
```

```
HEN POKE VARPTR(A$),PEEK(VARPTR(A$))-1:LE=LE+1:GOTO 16000
16300 IF ASC(B$)<33 OR B$="." THEN GOTO 16000
16400 IF LE>0 THEN A$=A$+B$:LE=LE-1
16500 GOTO 16000
16600 FOR QU=1 TO 10:NEXT QU
16700 SCREEN 0,1
16800 RETURN
16900 FOR SS=384 TO 448 STEP 32:PRINT @ SS,STRING$(32,32);:NEXT SS
17000 PRINT @ 480,STRING$(31,32);:POKE 1535,96
17100 GOSUB 16600
17200 RETURN
17300 Z$=STRING$(3-LEN(STR$(A)),48)+RIGHT$(STR$(A),LEN(STR$(A))-1):RETURN
17400 SOUND 1,1:A=C:GOSUB 17300:PRINT@DA(C,0),Z$;:PRINTTR$(DA(C,1));:A=DA(C,2):G
OSUB 17300:PRINTZ$;:GOSUB 16600:SOUND 250,1:RETURN
17500 GOSUB 16600:D=1
17600 DIM DA(16,11)
17700 FOR A=1 TO 16:READ DA(A,0):NEXT A
17800 FOR A=1 TO 16:B=2
17900 B=B+1:READ DA(A,B):SOUND A*10,1:IF DA(A,B)<>0 THEN GOTO 17900 ELSE NEXT A
18000 P$="T25;O3;1;3;5;6;8;10;12;O4;1;1;O3;12;10;8;6;5;3;1"
18100 GR=2:FOR W=1 TO SP:ZU(W)=0:NEXT W
18200 RETURN
18300 DATA 33,39,174,212,78,84,90,343,161,167,225,231,270,334,328,321
18400 DATA 9,2,0,1,3,5,9,10,0,2,4,5,6,10,12,13,0,3,5,6,7,8,13,14,0,2,3,6,10,0,3,
4,5,7,0,4,6,8,0,4,7,14,0,1,2,10,11,12,0,2,3,9,11,12,13,0,9,10,12,15,16,0,3,9,10,
11,13,14,15,16,0,3,4,10,12,14,15,0,4,8,12,13,15,0,11,12,13,14,16,0,11,12,15,0
```



ZX-81

Draw

ZX-DRAW

Dieses Zeichenprogramm für den I6K ZX--8I weist einige Besonderheiten auf.

Sie können bis zu 9 Bilder auf Kassette abspeichern und jederzeit wieder abrufen. Auch ein Vervollständigen der Bilder zu einem späteren Zeitpunkt ist möglich.

Die Bildschirmdarstellung ist 24zeilig. Die beiden unteren Zeilen enthalten Bedienungshinweise oder tragen Bildunterschriften (Infofeld).

Eine spezielle Maschinencoderoutine ermöglicht es, 24zeilige Hardkopien der Bilder mit dem ZX-Drucker anzufertigen. Eine andere Routine zeichnet nach Eingabe eines Radius Kreise an jede beliebige Stelle des Bildschirms.

DIE EINGABE

Geben Sie das Programm ein. A$ in Zeile IØ kann auch bescheidener ausfallen (z. B. IØ LET A$ = „ZX - DRAW“). Zeile eins ist für den 24zeiligen Ausdruck mit dem Drucker wichtig. Sie wird direkt eingegeben. Bei den Graphikzeichen handelt es sich um das Graphik-„A“-Zeichen. Vergessen Sie nicht, zur Sicherheit zwischendurch abzusaven. Durch die hohe Dimensionierung dauert das Absaven normalerweise recht lange. Wenn **keine** Bilder mitabgespeichert

werden sollen, empfiehlt sich die direkte Eingabe von „Break", „Clear" und Save „ZX DRAW". Die Abspeicherzeit verkürzt sich drastisch. Das Programm muß dann aber nach dem Laden mit „RUN" neu gestartet werden.

Nach dem Starten des Programms mit „RUN" bietet eine Menue folgende Auswahl an (Bild 1):

1. Anleitung
2. Zeichnen
3. Bilderverzeichnis
4. Bilderausdruck
5. Bilder löschen
6. Abspeichern

Ein invertierter Charakter steht imme für eine Inkey$-Eingabe. Nur in Teil drei und fünf des Menues müssen Eingaben mit „NEWLINE" quittiert werden.

Die Routine im einzelnen:

1. Anleitung: Die Anleitung enthält kurze Bedienungshinweise zum Zeichenvorgang.

2. Zeichnen: Beim Anlaufen dieses Unterprogramms zeigt sich über dem Infofeld ein blinkender Punkt, der sich mit den Pfeiltasten 5, 6, 7 und 8 über den Bildschirm steuern läßt. Die Taste „1" läßt den Punkt permanent erscheinen und ermöglicht damit den Zeichenvorgang. Die Taste „Ø" läßt den Punkt blinken und erlaubt beispielsweise das Radieren.

Mit der Taste „P" kehrt der „Zeichenstift" zu seiner Ausgangsposition zurück.

Das Infofeld bietet im Zeichenmodus drei Möglichkeiten.

I. Nach der Beendigung des Zeichenvorganges kann ein Bild mit der Taste „A" intern in das RAM abgespeichert werden. Der Computer fragt vorher an, unter welcher Nummer er das Bild abspeichern soll.

Die Ziffern eins bis neun sind möglich. Unter einer Nummer kann jeweils nur ein Bild abgespeichert werden. Später kann als Merkhilfe die Nummer sowie ein dazugehöriger Name in das Bilderverzeichnis eingetragen werden.

II. Durch Anwählen des Buchstaben „K" läßt sich um den Zeichenstift ein Kreis mit beliebigem Radius schlagen. Achten Sie darauf, daß der Kreis nicht die Bildschirmgrenzen sprengt, sonst steigt das Programm mit einer Fehlermeldung aus. Ein Radius muß immer zweistellig eingegeben werden, z. B. „03" oder „11".

III. Der Sprung zum Menuè erfolgt mit der Taste „M". Wichtig daran ist, daß auch das Bild gelöscht wird, wenn es vorher nicht abgespeichert wurde.

3. Bilderverzeichnis: Das Bilderverzeichnis erlaubt es, die Bilder und deren Namen in einer Liste zu notieren. Jeder Nummer kann ein Name mit bis zu zwölf Buchstaben zugeordnet werden. Im Eingabemodus „E" wird erst die Nummer und dann der Name eingetippt. Beide Eingaben müssen mit „NEWLINE" abgeschlossen werden.

Dann springt das Programm automatisch weiter zum Ausgabemodus und listet das Bildspeicherinhaltsverzeichnis auf.

Der Ausgabemodus kann mit „A" auch direkt angesteuert werden.

4. Bilderausdruck: Dieses Unterprogramm erlaubt es, eingespeicherte Bilder auszudrucken. Zunächst muß die Bildnummer eingegeben werden. Anschließend fragt der ZX-8I, ob ein Bildschirmausdruck „B" oder eine Hardcopy „H" gewünscht wird.

Beide, sowohl der Bildschirmausdruck als auch die Hardcopy enthalten in der untersten Zeile den Namen des Bildes, sofern er vorher im Bilderverzeichnis eingetragen worden ist. Der Bildname wird automatisch immer in die Mitte der letzten Zeile gesetzt, wenn er aus einem zusammenhängenden Wort besteht (Bild 2).

Soll ein Bild noch verschönert, erweitert oder korrigiert werden, so führt die Taste „2" in den Zeichenmodus, „M" steuert das Menue an.

Nach Ausdruck der Hardcopy springt das Programm von allein zum Menue zurück.

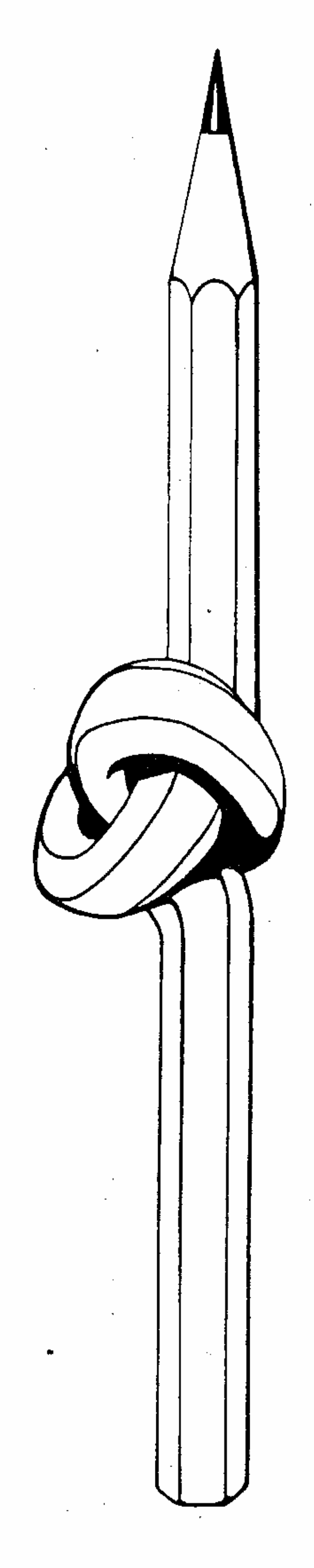
Selbstverständlich muß das erweiterte Bild anschließend wieder neu abgespeichert werden.

5. Bilder löschen: Die Bilder sowie ihre Namen können durch Eintippen der Bildnummern einzeln gelöscht werden.

6. Abspeichern: Das Programm wird inklusive Bildspeicher auf Kassette gesaved (Savezeit ca. sechs bis sieben Minuten).

ERWEITERUNGEN

Für Erwiterungen läßt das Programm nicht allzuviel Raum, weil der Bildspeicher eine Menge Platz benötigt. Die Zeile 9ØØØ gibt Auskunft darüber, wieviel Speicherplatz noch frei ist.



ZX-81

Ein Unterprogramm zum Zeichnen von schrägen Linien wäre bestimmt eine nützliche Erweiterung. Die Wortlänge des Bilderinhaltsverzeichnisses könnte durch Ändern der Zahl zwölf in den Zeilen sechs und 483Ø leicht vergrößert werden.

Die Bildspeicherroutine ließe sich auch gut als Ausgangsbasis für ein Trickfilmprogramm verwenden.

Denkbar ist hier für eine Begrenzung des Bildschirms, so daß kleinere, aber dafür mehr als neun Bilder im Speicher Platz finden.

Die Zeichenroutine ermöglichte das Zeichnen und Kopieren der Einzelbilder.

In Schleifen könnten dann die Bilder vorwärts oder rückwärts nacheinander übereinander ausgedruckt werden und einen Bewegungseffekt hervorrufen!

DAS PROGRAMM

Das Hauptprogramm ist das Menue (IØØØ–I2IØ), von dem die verschiedenen Unterprogramme angesteuert werden. Die Unterprogramme sind mit REM-Zeilen markiert. Der 24zeilige Bildschirm wird durch „Poke I64I8,Ø" erzeugt. Allerdings sind dann keine „Input"-Eingaben mehr möglich, das ist auch der Grund, warum überwiegend die komplizierteren „Inkey$"-Eingaben Verwendung finden.

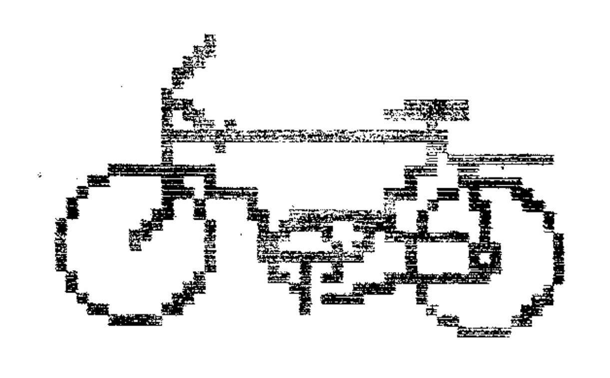
Erst „Poke I64I8,2" erzeugt wieder normale Verhältnisse.

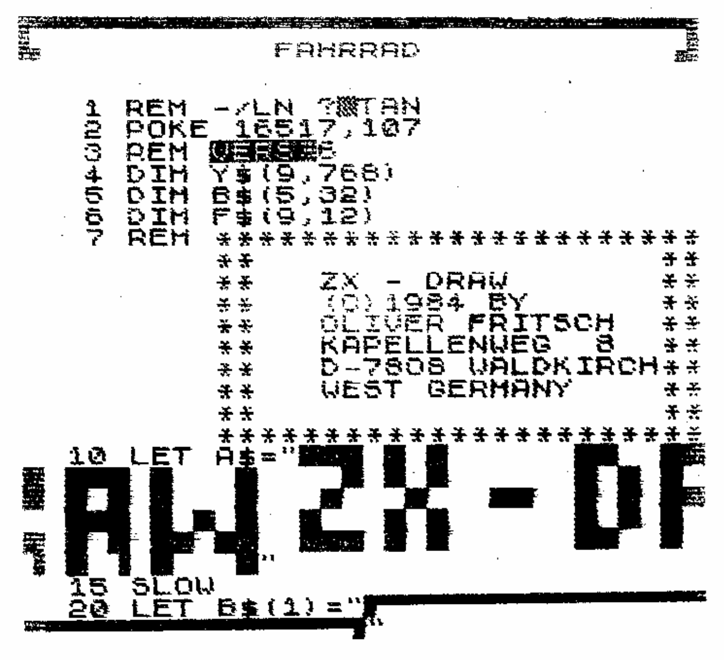
Der „COPY"-Befehl macht üblicherweise nur 22zeilige Bildschirmkopien, selbst dann, wenn 24 Zeilen auf dem Bildschirm zu sehen sind. Abhilfe schaffen hier Zeile eines und zwei. Der Befehl „RAND USR I65I4" ruft die kleine MC-Routine auf und der Drucker beginnt zu arbeiten.

Die Zeilen 3IØØ–3I4Ø tasten zeilenweise den Bildschirmspeicherinhalt ab und speichern ihn im String. Eine Bildschirmbegrenzung kann durch Verkleinern der Schleifen erreicht werden. Der String kann dann entsprechend kleiner dimensioniert sein.

ZX - DRAW

(C) 1984 BY OLIVER FRITSCH





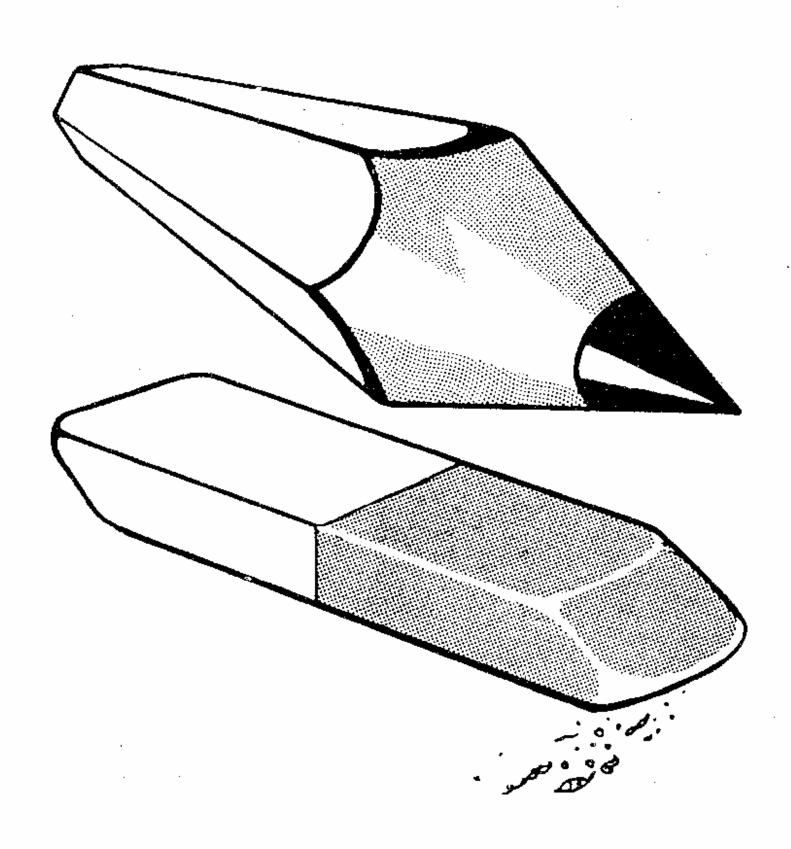
FAHRRAD

```
   1 REM -/LN ?▒TAN
   2 POKE 16517,107
   3 REM ██████
   4 DIM Y$(9,768)
   5 DIM B$(5,32)
   6 DIM F$(9,12)
   7 REM *************************
         **                     **
         **    ZX - DRAW        **
         **    (C) 1984 BY      **
         **    OLIVER FRITSCH   **
         **    KAPELLENWEG  8   **
         **    D-7808 WALDKIRCH **
         **    WEST GERMANY     **
         **                     **
         *************************
  10 LET A$="ZX - DRAW"
  15 SLOW
  20 LET B$(1)="▀▀▀▀▀▀▀▀"
```

```
  30 LET B$(2)="▌ █BSPEICHERN
█REIS     █ENUE ▟"
  40 LET B$(3)="▌                          █E
NUE
                 ▟"
  45 LET SW=0
  50 LET B$(4)="(C) 1984 BY OLIV
ER FRITSCH"
  60 FOR I=1 TO 9
  70 LET Y$(I)=" "
  80 NEXT I
 100 LET P=6
 110 LET FLAG=0
 500 REM █BILDSCHIRM█
 510 POKE 16418,0
 520 PRINT AT 4,1;A$
 530 PRINT AT 23,0;B$(3)
 540 PRINT AT 22,0;B$(1)
 550 PRINT AT 12,3;B$(4)
 560 PRINT AT 23,14;"M"
 570 IF INKEY$="M" THEN GOTO 100
0
 580 FOR I=1 TO 3
 590 NEXT I
 600 PRINT AT 23,14;"█"
 610 GOTO 530
1000 REM █MENUE█
1010 CLS
1020 POKE 16418,0
1030 PRINT AT 2,11;"M E N U E";A
T 3,11;"*********"
1040 PRINT AT 6,P;"█ ANLEITUNG"
1050 PRINT AT 8,P;"█ ZEICHNEN"
1060 PRINT AT 10,P;"█ BILDERVERZ
EICHNIS"
1070 PRINT AT 12,P;"█ BILDERAUSD
RUCK"
1080 PRINT AT 14,P;"█ BILDER LOE
SCHEN"
1090 PRINT AT 16,P;"█ ABSPEICHER
N"
1100 PRINT AT 22,0;B$(1)
1110 PRINT AT 23,0;"▌    NUMMER E
INGEBEN ( 1 - 6 )   ▟"
1130 LET M$=INKEY$
1140 PRINT AT 23,22;"█ - █"
1150 FOR I=1 TO 5
1160 NEXT I
1170 PRINT AT 23,22;"1 - 6"
1180 IF INKEY$="" THEN GOTO 1120
1190 IF CODE M$<29 OR CODE M$>34
 THEN GOTO 1130
1200 LET A=CODE M$-28
1210 GOTO A*1000+500
1500 REM █ANLEITUNG█
1510 CLS
1515 POKE 16418,0
1520 PRINT AT 0,1;A$
1530 PRINT ,,"DER ZEICHENSTIFT K
ANN MIT DEN   PFEILTASTEN ""5"",
""6"", ""7"" UND ""8"" UEBER DEN
BILDSCHIRM BEWEGT WER-DEN."
```

```
1540 PRINT ,,"DER STIFT LAESST S
ICH MIT DER   TASTE ""1"" AN- UN
D MIT ""0"" WIEDERABSETZEN."
1550 PRINT ,,"█ FUEHRT NACH DEM
BILDAUSDRUCK  WIEDER ZURUECK ZUM
 MENUE."
1555 PRINT "ABGESPEICHERTE BILDE
R KOENNEN   MIT █ VERVOLLSTAENDI
GT WERDEN."
1560 PRINT ,,"KREISRADIEN MUESSE
N ZWEISTELLIG EINGEGEBEN WERDEN,
Z.B. 02."
1750 PRINT AT 22,0;B$(1);AT 23,0
1800 PAUSE 4E4
1810 GOTO 1000
2500 REM ZEICHNEN
2510 CLS
2515 POKE 16418,0
2520 PRINT AT 23,0;B$(2);AT 22,0
;B$(1)
2530 LET C=5
2535 LET FLAG=0
2540 LET D=0
2550 LET D=D+(INKEY$="8" AND D<6
3)-(INKEY$="5" AND D>0)
2555 LET C=C+(INKEY$="7" AND C<4
3)-(INKEY$="6" AND C>0)
2587 IF INKEY$="1" THEN LET FLAG
=1
2588 IF INKEY$="0" THEN LET FLAG
=0
2590 PLOT D,C
2600 IF FLAG=0 THEN UNPLOT D,C
2610 IF INKEY$="M" THEN GOTO 100
0
2620 IF INKEY$="A" THEN GOTO 300
0
2630 IF INKEY$="K" THEN GOSUB 26
90
2680 GOTO 2550
2690 REM KREIS
2691 LET Y=0
2692 LET X=0
2693 LET V$=""
2701 PRINT AT 23,1;"RADIUS ZWEIS
TELLIG EING.ZB(04)"
2702 FOR I=1 TO 2
2703 LET O$=INKEY$
2704 IF INKEY$="" THEN GOTO 2703
2705 IF CODE O$>37 THEN GOTO 270
3
2706 LET V$=V$+O$
2707 PRINT AT 23,1;"          RAD
IUS: ";V$;"     ▟"
2708 FOR G=1 TO 8
2709 NEXT G
2711 NEXT I
2712 LET R=VAL V$
2713 IF R=0 THEN GOTO 2890
2715 FOR I=0 TO 2*PI STEP .1
2720 LET X=R*COS I+D
2730 LET Y=R*SIN I+C
2740 PLOT X,Y
2750 NEXT I
2760 PRINT AT 23,0;B$(2);AT 22,0
;B$(1)
2780 RETURN
3000 REM BILDSPEICHER
3010 PRINT AT 23,0;"▙     BILDNUM
MER ( 1 - 9 ) >█<  ▟"
3020 LET Q$=INKEY$
3030 IF INKEY$="" THEN GOTO 3010
3040 IF CODE Q$<29 OR CODE Q$>37
 THEN GOTO 3010
3050 LET Q=CODE Q$-28
3060 PRINT AT 23,27;Q
3070 FOR I=1 TO 10
3080 NEXT I
3090 FAST
3100 FOR W=1 TO 23
3110 FOR G=1 TO 32
3120 LET Y$(Q,G+32*W)=CHR$ PEEK
(PEEK 16396+256*PEEK 16397+G+33*
W)
3130 NEXT G
3140 NEXT W
3150 SLOW
3400 GOTO 2520
3500 REM BILDERVERZEICHNIS
3510 PRINT AT 23,2;"  █INGABE O
DER █USGABE ?"
3515 IF INKEY$="" THEN GOTO 3510
3520 IF INKEY$="E" THEN GOTO 360
0
3530 IF INKEY$="A" THEN GOTO 374
0
3540 GOTO 3510
3600 CLS
3610 POKE 16418,2
3620 PRINT "BITTE BILDNUMMER EIN
GEBEN (1-9)"
3630 PRINT AT 15,3;"...ANSCHLIES
SEND ""NEWLINE"" ";AT 19,11;"DRUE
CKEN"
3640 INPUT B
3645 IF B>9 OR B<1 THEN GOTO 360
0
3650 PRINT AT 4,10;"BILDNUMMER:
";B
3660 FOR I=1 TO 20
3670 NEXT I
3680 CLS
3690 PRINT "     BITTE BILDNAMEN
EINGEBEN."
3700 PRINT AT 4,5;"MAXIMAL 12 BU
CHSTABEN."
3710 INPUT F$(B)
3715 PRINT AT 6,5;"BILDNAME: ";F
$(B)
3720 FOR I=1 TO 20
3730 NEXT I
3740 POKE 16418,2
3750 CLS
3755 PRINT TAB 7;"BILDERVERZEICH
NIS";TAB 39;"-----------------"
3760 FOR I=1 TO 9
3770 PRINT AT I+3,4;I;" - ";F$(I
)
3780 NEXT I
3790 PRINT AT 20,0;B$(1);AT 21,0
;B$(3)
3800 PAUSE 4E4
3900 GOTO 1000
4500 REM BILDERAUSDRUCK
4505 LET SW=0
4510 PRINT AT 23,0;"▙     BILDNUM
MER ( 1 - 9 ) >█<  ▟"
4515 FOR I=1 TO 20
4517 NEXT I
4520 LET E$=INKEY$
4530 IF INKEY$="" THEN GOTO 4510
4540 IF CODE E$<29 OR CODE E$>37
 THEN GOTO 4510
4550 LET E=CODE E$-28
4560 PRINT AT 23,27;E
4570 FOR I=1 TO 20
4580 NEXT I
4590 PRINT AT 23,0;"▙ BILDSCHIRM
AUSDRUCK  █ARDCOPY ▟"
4595 IF INKEY$="" THEN GOTO 4595
4600 IF INKEY$="H" THEN GOSUB 46
50
4610 IF INKEY$="B" THEN GOSUB 48
00
4620 GOTO 4595
4650 LET SW=1
4800 CLS
4805 POKE 16418,0
4810 PRINT Y$(E)
4820 PRINT AT 22,0;B$(1);AT 23,0
;"▟"
4830 FOR I=1 TO 12
4840 IF NOT F$(E,I)=" " THEN NEX
T I
4850 PRINT AT 23,16-I/2;F$(E)
4860 IF SW=0 THEN GOTO 4900
4870 RAND USR 16514
4880 LET SW=0
4900 IF INKEY$="M" THEN GOTO 100
0
4902 IF INKEY$="Z" THEN GOTO 252
0
4904 GOTO 4900
```

```
4910 GOTO 1000
5500 REM B.LOESCHEN
5502 POKE 16418,2
5505 CLS
5510 PRINT TAB 8;"BILDER LOESCHE
N";TAB 40;"----------------"
5520 PRINT AT 20,0;B$(1);AT 21,0
;"▙ NUMMER DES BILDES EINGEBEN.
▟"
5530 INPUT B
5550 LET Y$(B)=""
5555 LET F$(B)=""
5560 PRINT AT 21,2;"  BILD NUMME
R ";B;" GELOESCHT. "
5570 FOR I=1 TO 60
5580 NEXT I
5590 PRINT AT 21,0;B$(3)
5800 PAUSE 4E4
5900 GOTO 1000
6500 REM ABSPEICHERN
6510 CLS
6520 PRINT AT 4,1;"KASSETTENRECO
RDER EINSCHALTEN,"
6530 PRINT AT 16,3;"...ANSCHLIES
SEND ""NEWLINE"" "
6540 PRINT AT 19,11;"DRUECKEN."
6550 PAUSE 4E4
6560 SAVE "ZX - DRA█"
6570 CLS
6580 GOTO 1020
9000 PRINT PEEK 16386-PEEK 16412
+256*(PEEK 16387-PEEK 16413)-50
2505 IF INKEY$="" THEN GOTO 253
0
```



Mooncrash

MOONCRASH

Ein nicht unrealistisches Simulationsspiel mit psychologischem Hintergrund für den ZX-81 mit 16 K Speichererweiterung, bei dem es außer auf schnelles Reaktionsvermögen auch auf gute Kombinationsgabe und Voraussicht ankommt.

Es liegt folgende Ausgangsposition vor: Sie sind ein Raumfahrer und Ihre Mondlandefähre ist über dem Mond abgestürzt. Übriggeblieben ist Ihnen nur Ihr solarzellenbetriebener Mondbuggy mit einem begrenzten Sauerstoffvorrat.

Ihre Aufgabe ist es jetzt, die sichere Mondbasis zu erreichen, bevor der Sauerstoffvorrat zur Neige geht.

Der vorhandene Sauerstoffvorrat nimmt konstant ab. Anders sieht es mit der Batterie aus. Ihre Kapazität ist zu gering, um mit einer Ladung die Mondbasis zu erreichen.

Konsequenterweise wird es notwendig sein, einige Zwischenstopps einzulegen, um die Batterie mittels der Solarzellen wieder aufzuladen.

Wie der Sand in den Wüsten dieser Erde, verschlingt auch die Mondoberfläche nur wenig Energie bei langsamer Fahrt, dafür aber überproportional viel Energie bei schneller Fahrt.

Wer also in einem Anfall von Raumkoller versucht, wie ein Wilder in die rettende Basis zu rasen, wird öfter zu einem Halt gezwungen werden, als ein gemächlicher Fahrer.

Wer wiederum zu langsam fährt, wird Schwierigkeiten mit seinem Luftvorrat bekommen.

Als weitere Schwierigkeit tauchen von Zeit zu Zeit Felsbrocken auf dem Fahrtweg auf. Diese Hindernisse lassen sich nur mit einem mechanischen Schutzschild aus dem Wege räumen. Je höher die Geschwindigkeit ist, desto kürzer muß die Reaktionszeit sein, wenn man nicht auf ein Hindernis knallen will und sich nicht seiner letzten Chance berauben will, die Mondbasis zu erreichen. Bonuspunkte auf den Sauerstoffvorrat in Abhängigkeit der gefahrenen Geschwindigkeit gibt es bei erfolgreicher Hindernisbewältigung.

Der Bildschirmaufbau

Der Bildschirm ist horizontal in zwei Hälften aufgeteilt. Die obere Hälfte zeigt die Kontrollinstrumente des Mondautos. Sauerstoffvorrat (Oxygen), Geschwindigkeit (Gang), Ladung und Distanz werden laufend angezeigt. Eine besondere Anzeige leuchtet bei Gefahr (Felsbrocken) auf. Wenn die Batterie aufgeladen wird, zeigt ein Pluszeichen („+") den Ladevorgang an. Sinkt der Luftvorrat unter zehn Einheiten ab, macht ein blinkender Pfeil auf die kritische Situation aufmerksam.

Die untere Hälfte des Bildschirms zeigt in einer Seitenansicht die Mondbasis, die Mondoberfläche sowie den Mondbuggy (Bild I).

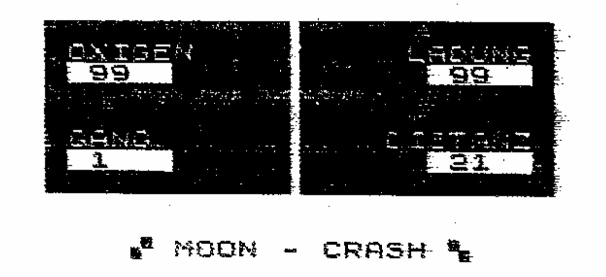
Die Bedienung

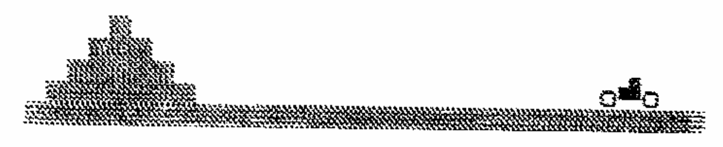
Das Spiel läßt sich durch Betätigen der Taste „S“ starten, wenn die Schrift auf dem Bildschirm dazu auffordert.

Ebenfalls mit der Taste „S“ läßt sich bei Gefahr der Schutzschild bedienen. Die Geschwindigkeit wird mit den Tasten 0-5 geregelt.

Nach dem Aufladen der Batterie muß das Fahrzeug wieder neu gestartet werden, d. h. ein gleichzeitiges Aufladen **und** Fahren ist nicht möglich. Das Spiel ist beendet nach einem Zusammenstoß mit einem Felsbrocken, wenn der Sauerstoffvorrat zu Ende ist oder wenn sich das Tor der Mondbasis ganz hinter dem Fahrzeug geschlossen hat.

Nach Drücken einer beliebigen Taste und der Taste „S“ kann das Spiel wieder neu gestartet werden.





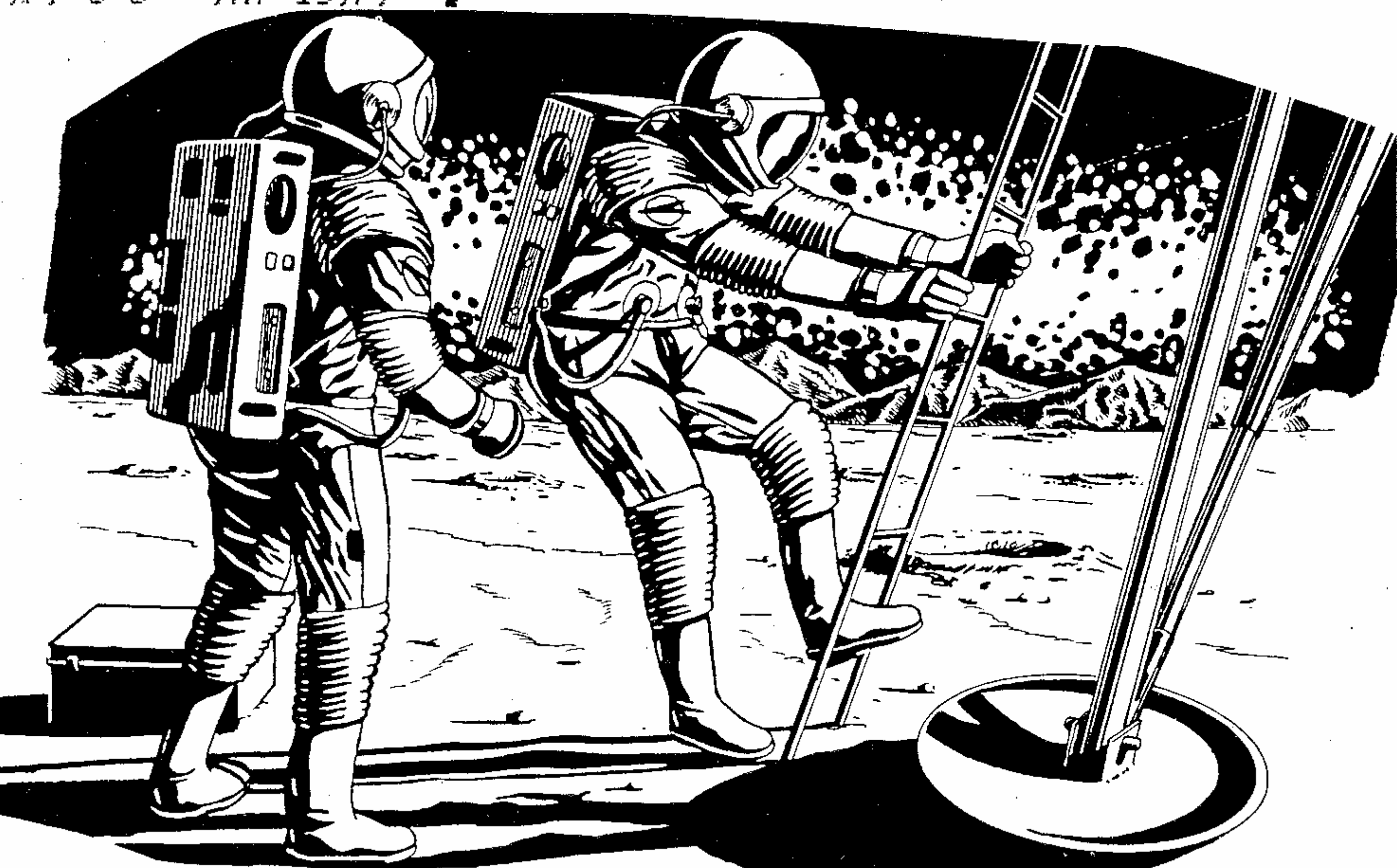
(BILD 1)

```
   5 REM          MOONCRASH
  10 REM ************************
  20 REM **                    **
  30 REM ** (C) 1984 BY        **
  40 REM ** OLIVER FRITSCH     **
  50 REM ** KAPELLENWEG 8      **
  60 REM ** D-7808 WALDKIRCH 2 **
  70 REM ** VERS. (GER)        **
  75 REM **                    **
  80 REM ************************
  90 REM 
  99 FAST
 100 CLS
 110 LET A$="                                "
 120 LET B$="                                "
 130 LET C$="                                "
 140 LET D$="OXIGEN"
 150 LET E$="LADUNG"
 160 LET F$="GANG"
 170 LET G$="DISTANZ"
 180 LET H$="                                "
 190 LET M$="                                "
 200 LET K$="     "
 210 LET L$="                                R  ."
 220 LET CC=99
 230 LET DD=22
 300 REM RND
 310 LET ME=INT (RND*12)+10
 500 REM DISPLAY
 501 FOR I=1 TO 8
 510 PRINT A$
 520 NEXT I
 530 PRINT AT 21,0;B$
 540 PRINT AT 17,2;C$
 550 PRINT AT 1,5;D$;AT 1,21;E$
 560 PRINT AT 5,5;F$;AT 5,20;G$
 570 PRINT AT 12,0;H$
 580 PRINT AT 2,5;K$;AT 2,22;K$
 590 PRINT AT 6,5;K$;AT 6,22;K$
 600 FOR I=0 TO 7
 610 PRINT AT I,15;"█"
 620 NEXT I
 640 PRINT AT 2,23;CC
 645 LET S=1
 650 PRINT AT 6,6;S
 655 LET O=99
 660 PRINT AT 2,6;O
 670 PRINT AT 6,23;DD
 680 LET P=27
 690 LET D=.05
 700 LET C=2
 900 SLOW
 903 PRINT AT 10,3;M$
 905 PRINT AT 10,8;"  MOON - CRASH  "
 915 FOR I=1 TO 50
 920 NEXT I
 930 PRINT AT 10,5;"DRUECKE ""S"" FUER START"
 940 FOR I=1 TO 50
 945 IF INKEY$="S" THEN GOTO 1000
 950 NEXT I
 960 PRINT AT 10,3;"(C) 1984 BY OLIVER FRITSCH"
 965 FOR I=1 TO 50
 970 NEXT I
 980 GOTO 903
1000 REM OXIGENKONTROLLE
1005 PRINT AT 10,3;M$
1010 LET O=O-1
1030 GOSUB 2010
1035 PRINT AT 2,6;O
1060 GOSUB 3000
1070 GOSUB 4000
1100 GOTO 1010
2000 REM SPEEDCHECK
2010 IF INKEY$="0" THEN GOTO 2500
2020 IF INKEY$="1" THEN GOTO 2600
2030 IF INKEY$="2" THEN GOTO 2650
2040 IF INKEY$="3" THEN GOTO 2700
2050 IF INKEY$="4" THEN GOTO 2750
2060 IF INKEY$="5" THEN GOTO 2800
2100 RETURN
2500 REM RECHARGE
2505 LET S=0
2507 PRINT AT 6,6;S
2508 PRINT AT 2,24;"+"
2510 IF NOT CC+3.5>99 THEN LET CC=CC+3.5
```

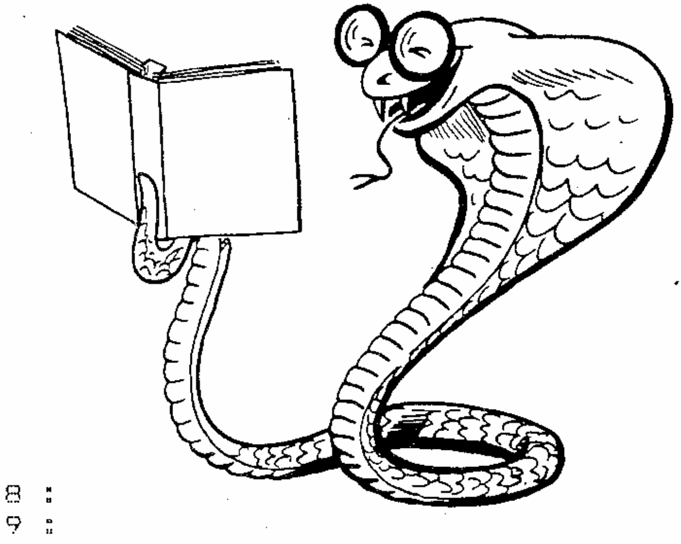
```
2520 PRINT AT 2,23;INT (CC)
2530 IF NOT INKEY$="" THEN GOTO
2010
2540 LET O=O-1
2560 PRINT AT 2,6;O
2565 GOSUB 3000
2570 GOTO 2510
2599 REM MEMORY
2600 LET S=1
2610 LET D=.1
2620 LET C=2
2640 GOTO 2900
2650 LET S=2
2660 LET D=.15
2670 LET C=4
2690 GOTO 2900
2700 LET S=3
2710 LET D=.4
2720 LET C=7
2740 GOTO 2900
2750 LET S=4
2760 LET D=.7
2770 LET C=11
2790 GOTO 2900
2800 LET S=5
2810 LET D=1.1
2820 LET C=16
2910 PRINT AT 6,6;S
2950 GOTO 1035
2999 REM DISP. CORRECTION
3000 IF O<=9 THEN PRINT AT 2,7;"
<"
3010 IF O<=9 THEN PRINT AT 2,7;"
 "
3020 IF O=0 THEN GOTO 9000
3030 RETURN
4000 REM CHARGE
4010 IF CC-C<=0 THEN GOTO 2500
4020 LET CC=CC-C
4030 PRINT AT 2,23;INT (CC)
4040 IF CC<=9 THEN PRINT AT 2,24
;"<";AT 2,24;" "
4100 LET P=P-D
4500 REM GRAPHICS
4501 IF NOT P<=3 THEN PRINT AT 2
0,P;"O▀O  ";AT 19,P;" ▪  "
```

```
4510 IF P<=3 THEN GOTO 8200
4600 REM DISTANCE
4602 IF NOT DD-D<=0 THEN LET DD=
DD-D
4610 PRINT AT 6,23;INT (DD)
4630 IF INT (DD)=ME THEN GOSUB 5
000
4640 IF DD<=10 THEN PRINT AT 6,2
4;"<";AT 6,24;" "
4790 GOTO 1000
5000 REM METEORS
5020 LET MET=P-1
5030 PRINT AT 20,MET;"*"
5040 PRINT AT 10,0;L$
5050 FOR I=1 TO 62 STEP C
5070 IF INKEY$="5" THEN GOTO 510
0
5080 NEXT I
5083 PRINT AT 10,1;M$
5085 FOR I=1 TO 20
5090 PRINT AT 10,11;"* CRASH * "
5092 PRINT AT 10,11;"█ CRASH █"
5093 NEXT I
5095 GOTO 9000
5100 PRINT AT 10,3;M$
5105 PRINT AT 20,MET;"("
5107 LET ME=INT (RND*12)+10
5108 LET O=O+5
5110 GOTO 2500
5200 RETURN
6000 STOP
8200 PRINT AT 20,7;"▚";AT 19,6;"
*"
8205 PRINT AT 6,23;"0"
8210 PRINT AT 10,5;"GRATULIERE,
GESCHAFFT."
8220 PAUSE 4E4
8230 RUN
9000 PRINT AT 10,11;" VERLOREN."
9010 GOTO 8220
9990 SAVE "MOONCRAS█"
9999 RUN

4606 IF S=0 THEN PRINT AT 6,6;"
```



Snake



SNAKE
Ein schnelles Spiel gegen den Apple II.
Auf einer begrenzten Fläche befinden sich zwei Schlangen. Sie sind die rosa Schlange, der Computer die blaue. Beide Schlangen bewegen sich ständig, wobei sie jedesmal um einen Punkt länger werden. Der freie Raum wird immer weniger, wodurch die Überlebenschancen der Schlangen sinken: denn, wenn sie gegen eine Mauer oder gegen die andere Schlange laufen, sind sie tot.
Sie kontrollieren Ihre Schlange über die Taste ‚A' (hoch), ‚Z' (abwärts), Pfeil rechts (rechts) und Pfeil links (links). Der Computer steuert seine Schlange und versucht sein möglichstes, Sie in die Enge zu treiben.
Vor jedem Spiel haben Sie die Möglichkeit, zwischen Standard- und zufälligen Anfangspositionen zu wählen.
Viel Spaß!

```
8 :
9 :
10  REM ************************
11  REM *                      *
12  REM *        SNAKE         *
13  REM *                      *
14  REM * (C) ROBERT TOLKSDORF *
15  REM *     AUGASSE 25       *
16  REM * 875 ASCHAFFENBURG    *
17  REM *                      *
18  REM *                      *
19  REM *                      *
20  REM *  LENGTH : 2.3 KBYTE  *
21  REM *                      *
22  REM ************************
23 :
24 :
40  HOME
50  FOR I = 768 TO 796: READ A: POKE I,A: NEXT
60  DATA 162,50,138,168,136,32,28,3,32,28,3,32,28,3,32,28,3,208,241,141,48
    ,192,232,224,250,144,231,96,96
70  DATA 0,-1,1,0,0,1,-1,0
80  FOR I = NU TO 3: READ AX(I),AY(I): NEXT
90 D$(0) = "A":D$(1) =  CHR$ (21):D$(2) = "Z":D$(3) =  CHR$ (8)
100 Z1 = 10:Z2 = 18:Z3 = 25
110 NU = 0:EI = 1:ZW = 2
120 CP = 1:CC = 2
130 SP =  - 16336
140 S1$ = "****************************************"
150 S2$ = "*                                      *"
160  GOTO 520
170  GR : POKE  - 16302,0: CALL  - 1998
180  COLOR= 10: HLIN 0,39 AT 0: VLIN 0,47 AT 39: HLIN 39,0 AT 47: VLIN 47,
     0 AT 0
190  HLIN 1,38 AT 1: VLIN 1,46 AT 38: HLIN 38,1 AT 46: VLIN 46,1 AT 1
200 PX = 17:PY = 20:DP = NU
210 CX = 23:CY = 20:DC = 2
220  IF RN THEN 250
230 PX = Z1 +  INT ( RND (EI) * Z2):PY = Z1 +  INT ( RND (EI) * Z3):DP =  INT
     ( RND (EI) * 4)
240 CX = Z1 +  INT ( RND (EI) * Z2):CY = Z1 +  INT ( RND (EI) * Z3):DC =  INT
     ( RND (EI) * 4)
```

```
250  IF  PEEK ( - 16384) < 128 THEN 290
260  GET A$
270  FOR I = NU TO 3: IF A$ = D$(I) THEN DP = I:I = 3
280  NEXT
290 PX = PX + AX(DP):PY = PY + AY(DP)
300  IF  SCRN( PX,PY) > NU THEN ER = NU: GOTO 470
310  COLOR= CP: PLOT PX,PY:DD =  PEEK (SP)
320  IF  SCRN( CX + AX(DC),CY + AY(DC)) = NU AND  SCRN( CX + AX(DC) * ZW,C
     Y + AY(DC) * ZW) = NU THEN 430
330 AD = DC + 1: IF AD > 3 THEN AD = AD - 4
340 A1 = ( SCRN( CX + AX(AD),CY + AY(AD)) = NU)
350  IF A1 THEN A1 = A1 + ( SCRN( CX + AX(AD) * ZW,CY + AY(AD) * ZW) = NU)

360 BD = AD + 2: IF BD > 3 THEN BD = BD - 4
370 B1 = ( SCRN( CX + AX(BD),CY + AY(BD)) = NU)
380  IF B1 THEN B1 = B1 + ( SCRN( CX + AX(BD) * ZW,CY + AY(BD) * ZW) = NU)

390  IF A1 = NU AND B1 = NU THEN 430
400  IF A1 > B1 THEN DC = AD
410  IF B1 > A1 THEN DC = BD
420  IF A1 = B1 THEN DC = AD: IF  RND (EI) > .5 THEN DC = BD
430 CX = CX + AX(DC):CY = CY + AY(DC)
440  IF  SCRN( CX,CY) > NU THEN ER = 1: GOTO 470
450  COLOR= CC: PLOT CX,CY:DD =  PEEK (SP)
460  GOTO 250
470  COLOR= 15
480  IF ER THEN PP = PP + 1: PLOT CX,CY
490  IF  NOT ER THEN PC = PC + 1: PLOT PX,PY
500  CALL 768
510  FOR I = 1 TO 1000: NEXT
520  TEXT : HOME
530  POKE  - 16368,0
540  PRINT S1$
550  FOR I = 1 TO 22: PRINT S2$: NEXT
560  PRINT S1$;
570  VTAB 3: HTAB 11: INVERSE : PRINT "        SNAKE        ": NORMAL
580  VTAB 5: HTAB 11: PRINT "BY ROBERT TOLKSDORF"
590  VTAB 8: HTAB 5: PRINT "YOU CONTROL YOUR PINK      A"
600  VTAB 9: HTAB 5: PRINT "                          <-   ->"
610  VTAB 10: HTAB 5: PRINT "SNAKE BY THESE KEYS:       Z"
620  VTAB 13: HTAB 5: PRINT "YOU HAVE"
630  VTAB 15: HTAB 5: PRINT PP;
640  IF PP = 1 THEN  PRINT " GAME";
650  IF PP <  > 1 THEN  PRINT " GAMES";
660  PRINT " WON AND"
670  VTAB 17: HTAB 5: PRINT PC;
680  IF PC = 1 THEN  PRINT " GAME";
690  IF PC <  > 1 THEN  PRINT " GAMES";
700  PRINT " LOST SO FAR."
710  VTAB 20: HTAB 5: PRINT "PRESS 'S' FOR STANDARD STARTING"
720  VTAB 22: HTAB 3: PRINT "LOCATIONS, ANY OTHER KEY FOR RANDOM";
730  WAIT  - 16384,128
740  GET A$
750 RN = NU
760  IF A$ = "S" THEN RN = 1
770  GOTO 170
```

Super - Datei

Super-Datei

Das hier vorgestellte Apple-II-Programm verwaltet Daten aller Art. Beispielsweise können Sie Ihre Schallplatten, Videofilme, Programme, Schauspieler, Adressen und vieles mehr durch den Computer verwalten lassen. Jeder Datensatz kann bis zu 15 verschiedene Informationen aufnehmen. Die unterschiedlichen Kriterien werden in Data's eingetragen.
Beispiel für ein Videofilmverzeichnis:
60 000 Data Filmtitel, Länge, Regie, Hauptdarsteller, Bemerkung.
Beispiel für eine Programmsammlung:
60 000 Data Programmname, Länge, Programmiersprache, Autor, Entstehungsjahr, Bemerkung.
Am Ende der Data-Zeile muß ein Komma stehen, da sonst eine Fehler meldung ausgegeben wird.

Hat man die Data-Zeile eingegeben, so kann man das Programm mit RUN starten. Es stehen nun folgende Befehle zur Verfügung:

(S) Suchen
(A) Daten ausgeben
(N) Neuer Eintrag
(O) Alphabetisch sortieren
(Q) Daten abspeichern und Programm beenden

Durch den Befehl (N) können Daten eingetragen werden. Durch (S) kann ein bestimmtes Kriterium nach einem Stichwort abgesucht werden. Wird dieses gefunden, so wird der gesamte Datensatz angezeigt und man kann diesen nun Editieren, Löschen oder auf Drucker ausgeben lassen. Durch (O) können die Datensätze nach einem beliebigen Kriterium sortiert werden. Mit dem Befehl (A) kann ein beliebiges Kriterium auf Drucker oder Bildschirm ausgegeben werden.

Sind alle Eingaben oder Änderungen vorgenommen, so werden die Daten sowie das Datum mittels (Q) auf Diskette gespeichert.
Nach dem Starten des Programms werden die Daten immer automatisch eingelesen.

Das Programm arbeitet weitgehend mit einem Hauptmenü, so daß weitere Erläuterungen entfallen können.
Die Anzahl der Datensätze richtet sich nach Speicherkapazität, Anzahl der Kriterien, Länge der Daten.

```
10  REM  ********************
20  REM   SUPER- D A T E I
30  REM   (C) F.BRALL 1984
40  REM  ********************
50  DIM S$(15),DD$(500,15),DP$(15)
60  ONERR  GOTO 1630
70 U = 0
80  READ S$(U)
90  IF S$(U) <  > "" THEN U = U + 1: GOTO 80
100  GOSUB 1470
110  HOME : INVERSE : PRINT "  SUPER-DATEI       (C) F.BRALL  1984   "
120  NORMAL
130  FOR I = 0 TO U - 1
140  PRINT "<"I">  " TAB( 7)S$(I)
150  NEXT I
160  VTAB (19)
170  PRINT "<S> SUCHEN          <A> DATEN AUSGEBEN"
180  PRINT "<N> NEUER EINTRAG   <Q> SAVE DATEN"
190  HTAB (20): PRINT "<O> SORTIEREN"
200  PRINT : PRINT "BEFEHL ->              DATUM:"DA$;: HTAB 10
210  GET E$
220  IF E$ = "N" THEN 290
230  IF E$ = "S" THEN 370
240  IF E$ = "Q" THEN 1330
250  IF E$ = "A" THEN 540
260  IF E$ = "O" THEN 820
270  GOTO 210
280  REM  *** NEUE EINTRAGUNG ***
290  HOME : PRINT "**** NEUE EINTRAGUNG ****"
300  PRINT
```

```
310  FOR I = 0 TO U - 1
320  PRINT  LEFT$ (S$(I),16); TAB( 17);
330  INPUT ": ";DD$(O,I)
340  NEXT I
350 O = O + 1: GOTO 110
360  REM  *** SUCHEN ***
370  VTAB 23: HTAB 1: INVERSE : PRINT "SUCHEN";
380  NORMAL : PRINT " NACH WELCHEM KRITERIUM ?        ";
390  HTAB 33: INPUT "";E$
400  IF  VAL (E$) < 0 OR  VAL (E$) > U - 1 THEN 370
410  VTAB 23: HTAB 1: PRINT "                                        ";
420  VTAB 23: HTAB 1:I =  VAL (E$):  PRINT S$(I)" ";
430  INPUT "? ";SW$
440  GOSUB 1100
450  IF FI < 0 THEN  FLASH : VTAB 23: HTAB 1: PRINT "   N I C H T  G E F U
      N D E N           ";: GET E$: GOTO 110
460  GOSUB 1160
470  VTAB 19: PRINT "<E> EDIT   <L> LOESCHEN   <C> HARTCOPY"
480  PRINT : PRINT  TAB( 15)"<SPACE> WEITER"
490  VTAB 23: PRINT "BEFEHL -->";
500  GET E$: IF E$ = "E" THEN  GOSUB 1240: GOTO 490
510  IF E$ = "C" THEN  HOME : PR# 1: PRINT : GOSUB 1180: PR# 0: CALL 1002:
      GOTO 460
520  IF E$ = "L" THEN 1570
530  GOTO 110
540  REM  DATEN AUSGEBEN
550  HOME : PRINT "*** DATEN AUSGEBEN ***"
560  PRINT
570  FOR I = 0 TO U - 1
580  PRINT "<"I">  " TAB( 7)S$(I)
590  NEXT I
600  PRINT : PRINT "<A>  ALLES AUSGEBEN"
610  VTAB 23: HTAB 1: PRINT "WELCHES KRITERIUM";
620  INPUT " ? ";E$
630  IF E$ = "A" THEN 650
640  IF  VAL (E$) > U - 1 THEN 550
650  VTAB 23: HTAB 1: PRINT "                                       "
660  VTAB 23: PRINT "MONITOR <M> ODER PRINTER <P>  -->";
670  GET OG$: IF OG$ <  > "M" AND OG$ <  > "P" THEN 650
680  IF OG$ = "P" THEN  PR# 1
690  IF E$ = "A" THEN 770
700  HOME
710 I =  VAL (E$):Q = 0
720  FOR T = 0 TO O - 1
730  PRINT DD$(T,I):Q = Q + 1
740  IF Q > 21 AND OG$ = "M" THEN Q = 0: VTAB 23: HTAB 2: PRINT "<SPACE>";
     : GET E$: HOME : PRINT
750  NEXT T
760  PRINT : PRINT "<SPACE>";: GET E$: GOTO 110
770  FOR FI = 0 TO O - 1
780  GOSUB 1160
790  IF OG$ = "M" THEN  VTAB 23: PRINT " <SPACE>";: GET E$
800  NEXT FI: PRINT : PR# 0: CALL 1002: GOTO 110
810  REM  SORTIEREN
820  VTAB 23: HTAB 1: PRINT "
830  INVERSE : VTAB 23: PRINT "SORTIEREN";: NORMAL
840  PRINT " NACH WELCHEM KRITERIUM";
850  INPUT " ? ";E$
860 T =  VAL (E$): IF T > U - 1 THEN 820
870  FOR L1 = 0 TO O - 2: FOR L2 = L1 + 1 TO O - 1
```

```
880  GOSUB 980
890  IF Q1 < Q2 THEN 910
900  GOSUB 930
910  NEXT : NEXT : GOTO 110
920  REM  UNTERPROGRAMME FUER BUBBLESORT
930  FOR I = 0 TO U - 1
940 DP$(I) = DD$(L1,I)
950 DD$(L1,I) = DD$(L2,I)
960 DD$(L2,I) = DP$(I)
970  NEXT I: RETURN
980 Z = 1
990 Q1$ =  MID$ (DD$(L1,T),Z,1)
1000 Q2$ =  MID$ (DD$(L2,T),Z,1)
1010  IF Z > 15 THEN 1030
1020  IF Q1$ = Q2$ THEN Z = Z + 1: GOTO 990
1030 Q1 = 0: IF Q1$ <  > "" THEN Q1 =  ASC (Q1$)
1040 Q2 = 0: IF Q2$ <  > "" THEN Q2 =  ASC (Q2$)
1050  RETURN
1060  REM  NACH STICHWORT SUCHEN
1070  REM  SW$=STICHWORT
1080  REM  I=KRITERIUM
1090  REM  FI=NR. DES EINTRAGES
1100 T = 0
1110  IF DD$(T,I) = SW$ THEN FI = T: RETURN
1120 T = T + 1: IF T > O - 1 THEN FI =  - 1: RETURN
1130  GOTO 1110
1140  REM  DATENSATZ ANZEIGEN
1150  REM  FI=NR. DES DATENSATZES
1160  HOME
1170  INVERSE : PRINT "
1180  NORMAL
1190  FOR I = 0 TO U - 1
1200  PRINT  LEFT$ (S$(I),16); TAB( 17)": "DD$(FI,I)
1210  NEXT I: RETURN
1220  REM  EDITIERE DATENSATZ
1230  REM  FI=NR. DES DATENSATZES
1240  VTAB 3: HTAB 1
1250  FOR I = 0 TO U - 1
1260  PRINT  LEFT$ (S$(I),16); TAB( 17);
1270 X =  PEEK (36):Y =  PEEK (37)
1280  INPUT ": ";E$: IF E$ <  > "" THEN DD$(FI,I) = E$
1290  VTAB (Y + 1): HTAB (X + 3): PRINT DD$(FI,I)
1300  NEXT I
1310  RETURN
1320  REM   SAVE DATEN
1330  HOME : VTAB 10: PRINT "DATUM: "DA$
1340  VTAB 10: HTAB 7: INPUT " ";E$: IF E$ <  > "" THEN DA$ = E$
1350  HOME : VTAB 10: PRINT "DATUM: "DA$
1360  PRINT : PRINT : PRINT "DATEN WERDEN GESPEICHERT !"
1370  PRINT  CHR$ (4)"OPEN DATEN"
1380  PRINT  CHR$ (4)"WRITE DATEN"
1390  PRINT O: PRINT DA$
1400  FOR I = 0 TO U - 1
1410  FOR T = 0 TO O - 1
1420  PRINT DD$(T,I)
1430  NEXT T,I
1440  PRINT  CHR$ (4)"CLOSE DATEN"
1450  END
1460  REM   LOAD DATEN
1470  PRINT  CHR$ (4)"OPEN DATEN"
```

```
1480  PRINT  CHR$ (4)"READ DATEN"
1490  INPUT O: INPUT DA$
1500  FOR I = 0 TO U - 1
1510  FOR T = 0 TO O - 1
1520  INPUT DD$(T,I)
1530  NEXT T,I
1540  PRINT  CHR$ (4)"CLOSE DATEN"
1550  RETURN
1560  REM  DATENSATZ LOESCHEN
1570 O = O - 1
1580  FOR T = FI TO O - 1
1590  FOR I = 0 TO U - 1
1600 DD$(T,I) = DD$(T + 1,I)
1610  NEXT I,T
1620  GOTO 110
1630  REM  ERROR UNTERDRUECKEN
1640 FE =  PEEK (222)
1650  IF FE = 5 THEN 110
1660  FLASH : PRINT "ERROR "FE: NORMAL : END
60000  DATA  FILMTITEL,LAENGE,REGIE,MUSIC,HAUPTDARSTELLER,BEMERKUNG,
```

Apple II

Shape-tables

Shape tables

Mit Hilfe dieses Programms können vom Benutzer bis zu 255 Hires-shapes mit einer Matrixgröße von 40x20 Pixels erstellt werden.
Bei Beginn wird gefragt, ob man die Startadresse $9000 beibehalten möchte. Dieser Bereich ist gut nutzbar, da viel Speicherkapazität zur Verfügung steht. Andere Adressen wären $2000-6000 im Bereich der HGR- und HGR2-Seite. Dabei muß man darauf achten, daß man die Programme nicht in der Seite schreibt, in die man die Shapes abgelegt hat. Sie gehen sonst verloren oder werden teilweise zerstört.
Hat man die Startadresse festgelegt, wird man nach der Anzahl der Shapes gefragt. Höchstens 255 und mindestens 1 Shape(s) müssen eingegeben werden. Für die Definition wrid eine Punktmatrix im Textfeld gezeigt, auf der man die Shapes definieren kann: I=nach oben, M=nach unten, J=nach links, K=nach rechts. Mit der Taste ‚A' werden Punkte gesetzt bzw. gelöscht. Im untenliegenden Feld werden die ‚X'- und ‚Y'-Werte der Cursorposition angegeben. Ist das Shape fertig, bewegt man den Cursor an die Stelle, wo man den Ausschnitt haben möchte. Angenommen, das Shape ist 12x5 Punte groß, bewegt man den Cursor an diese Stelle und drückt ‚RETURN'. Danach wird das Shape compiliert in den Hires-mode und ausgegeben, falls keine anderen Shapes definiert werden sollen. Möchte man das Shape abspeichern, gibt man den Namen ein und erhält automatisch die erste Programmzeile, die mit der Rechts-Pfeiltaste eingelesen wird.
Für Cassettenbenutzer: Die Dez.-Startadresse ist die Variable ‚G', die Endadresse errechnet sich einfach aus der Variablen ‚PO'+1. Die Hex.-Startadresse ist die Variable ‚SL$'.

```
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

Shape tables

(c) 1984 von Mark Heene

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

10  TEXT : HOME
20  GOSUB 930
30 S = S + 1: IF S > SH THEN 760
40  FOR I = 1 TO 20:L$(I) = "": NEXT I:C = 0: HOME
50  FOR Y = 1 TO 20: FOR X = 1 TO 40: HTAB X: VTAB Y: PRINT ".":P$(X,Y) =
    ".": NEXT X: NEXT Y
```

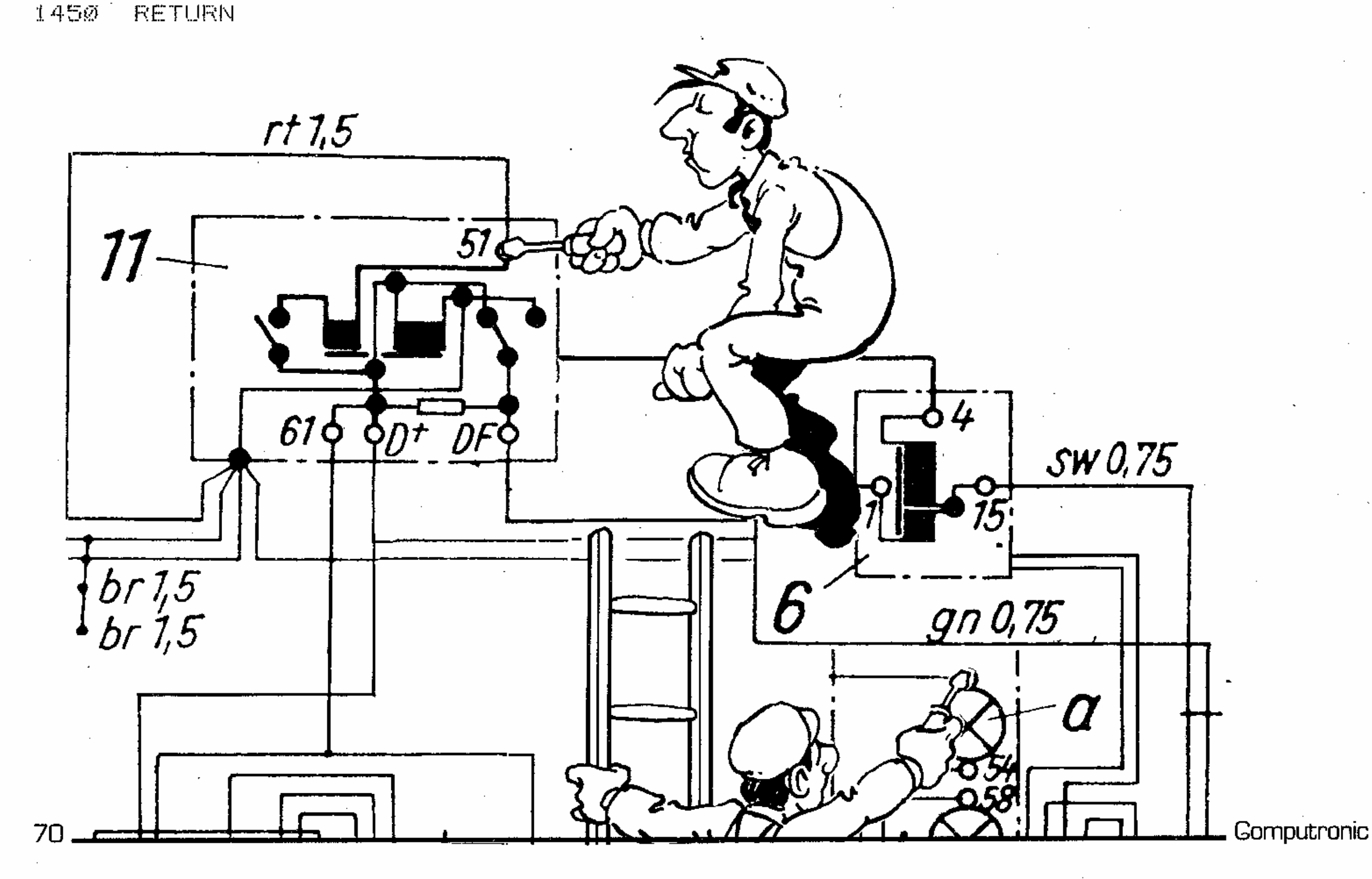
```
6Ø  POKE 34,2Ø
7Ø X = 1:Y = 1
8Ø  VTAB 22: HTAB 5: PRINT "X =     "
9Ø  VTAB 23: HTAB 5: PRINT "Y =     "
1ØØ  VTAB 21: HTAB 16: PRINT "SHAPE ";: INVERSE : PRINT S: NORMAL
11Ø  VTAB 23: HTAB 12: PRINT "<RETURN> BEENDET DEFINITION"
12Ø  VTAB 22: HTAB 9: PRINT "  ": VTAB 23: HTAB 9: PRINT "  ": VTAB 22: HTAB
     9: PRINT X: VTAB 23: HTAB 9: PRINT Y
13Ø  HTAB X: VTAB Y: GET A$
14Ø  IF A$ =  CHR$ (13) THEN 32Ø
15Ø  IF A$ = "I" THEN  GOSUB 21Ø: GOTO 2ØØ
16Ø  IF A$ = "M" THEN  GOSUB 23Ø: GOTO 2ØØ
17Ø  IF A$ = "J" THEN  GOSUB 25Ø: GOTO 2ØØ
18Ø  IF A$ = "K" THEN  GOSUB 27Ø: GOTO 2ØØ
19Ø  IF A$ = "A" THEN  GOSUB 29Ø: GOTO 2ØØ
2ØØ  GOTO 12Ø
21Ø Y = Y - 1: IF Y = Ø THEN Y = Y + 1: CALL  - 198
22Ø  RETURN
23Ø Y = Y + 1: IF Y = 21 THEN Y = Y - 1: CALL  - 198
24Ø  RETURN
25Ø X = X - 1: IF X = Ø THEN X = X + 1: CALL  - 198
26Ø  RETURN
27Ø X = X + 1: IF X = 41 THEN X = X - 1: CALL  - 198
28Ø  RETURN
29Ø  IF P$(X,Y) = "." THEN P$(X,Y) = "*": GOTO 31Ø
3ØØ P$(X,Y) = "."
31Ø  HTAB X: VTAB Y: PRINT P$(X,Y): RETURN
32Ø  HOME
33Ø  VTAB 21: HTAB 3: PRINT "X = "X", Y = "Y
34Ø X1 = 1
35Ø X2 = X
36Ø  IF X2 / 2 <  > INT (X2 / 2) THEN X2 = X2 + 1
37Ø Y1 = 1
38Ø Y2 = Y
39Ø  IF Y2 / 2 <  > INT (Y2 / 2) THEN Y2 = Y2 + 1
4ØØ  FOR I = Y1 TO Y2
41Ø C = C + 1
42Ø  FOR J = X1 TO X2
43Ø L$(C) = L$(C) + P$(J,I)
44Ø  NEXT J
45Ø  NEXT I
46Ø  TEXT : HOME : FOR I = 1 TO C: PRINT L$(I): NEXT I
47Ø  IF Q THEN 49Ø
48Ø  POKE G,SH: POKE G + 1,Ø:PO = G + SS + 1:P = G + 2:Q = 1
49Ø PP = PO - G + 1
5ØØ  GOSUB 121Ø
51Ø  POKE P,P1: POKE P + 1,P2:P = P + 2
52Ø  FOR I = 1 TO C STEP 2
53Ø  FOR J = X1 TO X2 - 1
54Ø C$ =  MID$ (L$(I),J,1)
55Ø  IF C$ = "." THEN KE = 1: GOTO 57Ø
56Ø KE = 5
57Ø PO = PO + 1: POKE PO,KE
58Ø  NEXT J
59Ø C$ =  MID$ (L$(I),J,1)
6ØØ  IF C$ = "." THEN KE = 2: GOTO 62Ø
61Ø KE = 6
62Ø PO = PO + 1: POKE PO,KE
63Ø  FOR J = X2 TO X1 + 1 STEP  - 1
64Ø C$ =  MID$ (L$(I + 1),J,1)
```

```
650  IF C$ = "." THEN KE = 3: GOTO 670
660 KE = 7
670 PO = PO + 1: POKE PO,KE
680  NEXT J
690 C$ =  MID$ (L$(I + 1),J,1)
700  IF C$ = "." THEN KE = 2: GOTO 720
710 KE = 6
720 PO = PO + 1: POKE PO,KE
730  NEXT I
740 PO = PO + 1: POKE PO,0
750  GOTO 30
760  TEXT : HOME : VTAB 12: HTAB 15: PRINT "FERTIG....": FOR I = 1 TO 1000
     : NEXT I
770  HOME : HGR
780 SX =  - 35:SY = 10: POKE 232,H1: POKE 233,H2
790  FOR I = 1 TO SH
800  HCOLOR= 3: ROT= 0: SCALE= 1
810 SX = SX + 45: IF SX > 240 THEN SX = 10:SY = SY + 25
820  DRAW I AT SX,SY
830  NEXT I
840  VTAB 23: HTAB 14: PRINT "TASTE..."
850  WAIT  - 16384,128: GET W$: TEXT : HOME
860  INVERSE : VTAB 5: HTAB 5: PRINT "SHAPES SPEICHERN ?";: NORMAL : INPUT
     "";Y$
870  IF Y$ = "N" THEN  HOME : END
880  VTAB 8: PRINT "DIE FOLGENDE PROGRAMMZEILE MIT DER       '->' TASTE EIN
     LESEN, UND ALS PROGRAMM-  ANFANG VERWENDEN."
890  VTAB 12: INPUT "NAME : ";N$
900  PRINT : PRINT  CHR$ (4)"BSAVE "N$".SHAPE ,A"G" ,L"PO - G + 1""
910  VTAB 15: PRINT "010 HIMEM:"G" : D$ = CHR$(4) : PRINT : PRINT D$" CHR$
     (34)"BLOAD "N$".SHAPE ,A"G"" CHR$ (34)" : POKE 232,"H1" : POKE 233,"H
     2""
920  VTAB 14: NEW
930  VTAB 5: INVERSE : PRINT "--- SHAPE TABLES 1984 VON MARK HEENE ---": NORMAL

940 SL$ = "9000"
950  VTAB 8: CALL  - 958: VTAB 8: HTAB 5: PRINT "START ADRESSE : $"SL$
960  VTAB 10: HTAB 5: INPUT "AENDERN ";Y$: IF Y$ = "N" OR Y$ = "" THEN 990

970  VTAB 10: HTAB 5: PRINT "NEUE ADRESSE (";: INVERSE : PRINT "FORMAT : X
     XXX";: NORMAL : PRINT ") $";: INPUT "";SL$
980  GOTO 950
990  DIM L$(20),P$(40,20)
1000  VTAB 15: HTAB 5: PRINT "ANZAHL DER SHAPES (MAX.255) : ";
1010  VTAB 15: HTAB 34: INPUT "";SH
1020  IF SH > 255 OR SH < 1 THEN  FLASH : VTAB 15: HTAB 24: PRINT "MAX.255
     ": NORMAL : GOTO 1010
1030  FOR I = 4 TO 1 STEP  - 1
1040 C$ =  MID$ (SL$,I,1)
1050  IF C$ >  = "A" AND C$ <  = "F" THEN D =  ASC (C$) - 55: GOTO 1070
1060 D =  VAL (C$)
1070 G = G + D * 16 ^ F
1080 F = F + 1: NEXT I:F = 0
1090  FOR I = 4 TO 3 STEP  - 1
1100 C$ =  MID$ (SL$,I,1)
1110  IF C$ >  = "A" AND C$ <  = "F" THEN D =  ASC (C$) - 55: GOTO 1130
1120 D =  VAL (C$)
1130 H1 = H1 + D * 16 ^ F:F = F + 1: NEXT I:F = 0
1140  FOR I = 2 TO 1 STEP  - 1
```

```
1150 C$ =  MID$ (SL$,I,1)
1160  IF C$ >  = "A" AND C$ <  = "F" THEN D =  ASC (C$) - 55: GOTO 1180
1170 D =  VAL (C$)
1180 H2 = H2 + D * 16 ^ F:F = F + 1: NEXT I:F = 0
1190 SS = 2 * SH
1200  HIMEM: G: RETURN
1210  DATA  3,768,2,512,1,256,F,240,E,224,D,208,C,192,B,176,A,160,9,144,8,
     128,7,112,6,96,5,80,4,64,3,48,2,32,1,16,F,15,E,14,D,13,C,12,B,11,A,10
     ,9,9,8,8,7,7,6,6,5,5,4,4,3,3,2,2,1,1
1220 P1 = 0:P2 = 0
1230 HX$ = "": RESTORE :PE = PP
1240  READ H$,DQ
1250  IF PE >  = DQ THEN PE = PE - DQ:HX$ = HX$ + H$
1260  IF PE = 0 THEN 1280
1270  GOTO 1240
1280  IF PP <  = 15 THEN HX$ = "000" + HX$: GOTO 1310
1290  IF PP <  = 256 THEN HX$ = "00" + HX$: GOTO 1310
1300  IF PP <  = 1000 THEN HX$ = "0" + HX$: GOTO 1310
1310  IF  LEN (HX$) = 3 THEN HX$ = HX$ + "0"
1320  IF  LEN (HX$) = 2 THEN HX$ = HX$ + "00"
1330 QQ = 0: FOR I = 4 TO 3 STEP  - 1
1340 C$ =  MID$ (HX$,I,1)
1350  IF C$ >  = "A" AND C$ <  = "F" THEN D =  ASC (C$) - 55: GOTO 1370
1360 D =  VAL (C$)
1370 P1 = P1 + D * 16 ^ QQ
1380 QQ = QQ + 1: NEXT I
1390 QQ = 0: FOR I = 2 TO 1 STEP  - 1
1400 C$ =  MID$ (HX$,I,1)
1410  IF C$ >  = "A" AND C$ <  = "F" THEN D =  ASC (C$) - 55: GOTO 1430
1420 D =  VAL (C$)
1430 P2 = P2 + D * 16 ^ QQ
1440 QQ = QQ + 1: NEXT I
1450  RETURN
```

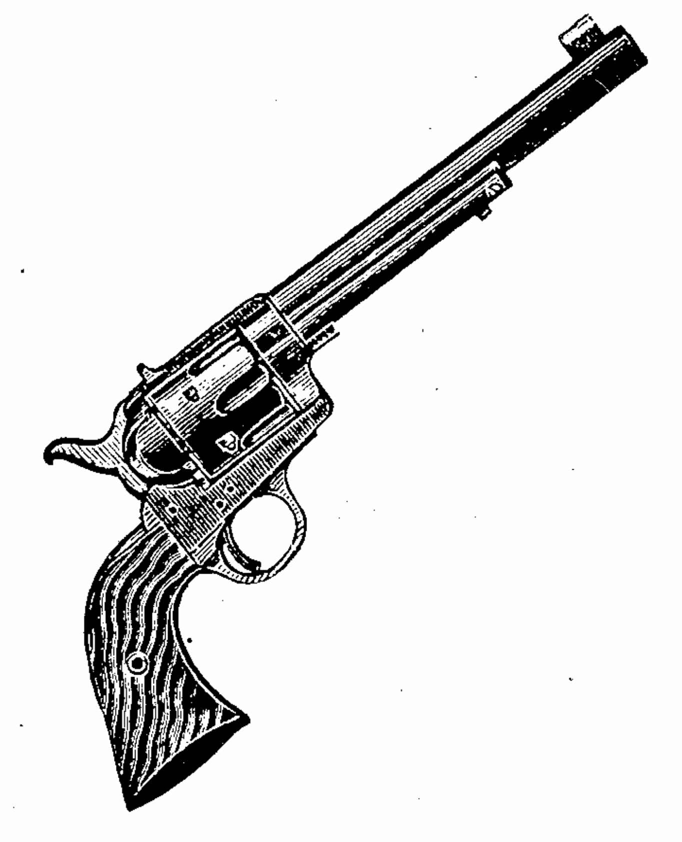


Tomstone - City

Tomstone-City

für Atari 600/800XL

In diesem Spiel muß man versuchen, durch Abschießen des roten Sheriffs soviel Punkte wie möglich zu erreichen. Dabei wird man von kleinen Männchen behindert, die Dir nachstellen und Dich vernichten wollen. Die kleine Stadt in der Mitte des Spielfeldes gewährt Dir Schutz vor diesen Feinden. Wenn alle Sheriffs vernichtet sind, gibt es einen Bonus – und ein neues Feld wird aufgebaut. Du kannst wählen zwischen 9 verschiedenen Schwierigkeitsgraden.





```
10 REM ***************************
11 REM *                         *
12 REM *      TOMBSTONE CITY     *
13 REM *                         *
14 REM *    (c) 11.4.1984 by     *
15 REM *                         *
16 REM *        Jens Berke       *
17 REM *       Finkenhof  7      *
18 REM *      3070  Nienburg     *
19 REM *    Tel.:05021/12077     *
20 REM *                         *
21 REM ***************************
22 REM
28 REM Z=Variable fuer LOCATE
29 REM SC=Score
30 REM ST=Stickabfrage
31 REM SO=Soundvariable
32 REM A,B,C,D=Vielzweckvariablen
33 REM H & V=Bewegungsrichtung Schooner
34 REM H1 & V1=Bewegungsrichtung Schuss
35 REM AF=Anzahl der Feinde
36 REM AK=Anzahl der Kakteen
37 REM AS=Anzahl der Sheriffs
38 REM FN=Feindnummer
39 REM KN=Kaktusnummer
40 REM KF=Flag fuer neuen Kaktus
41 REM FF=Flag fuer neuen Feind
42 REM LE=Anzahl der Leben
43 REM SG=Schwierigkeitsgrad
44 REM S(N)=COLOR fuer Schooner
45 REM FX & FY=Bewegungsrichtung Feind
46 REM SX & SY=Schoonerkoordinaten
47 REM PX & PY=Schusskoordinaten
48 REM FX(N) & FY(N)=Feindkoordinaten
49 REM KX(N) & KY(N)=Kaktuskoordinaten
50 REM
99 REM * Initialisierung *
```



```
100 DIM A$(44),KX(150),KY(150),S(15),FX(6),FY(6)
110 IF PEEK(16392)<>24 THEN POKE 559,0:GOSUB 32000
120 S(14)=161:S(13)=162:S(7)=163:S(11)=164:S(5)=163:S(6)=163:S(9)=164:S(10)=164
190 GOSUB 16000
200 GOSUB 30000:SC=0:LE=3
999 REM * Bildschirmaufbau *
1000 GRAPHICS 17:POKE 710,148:POKE 709,0:POKE 708,54:POKE 711,180:POKE 712,238:P
OKE 756,64
1010 GOSUB 31000:FN=0:KN=0:KX(0)=19:KY(0)=0:FF=0:KF=0:AF=0:AK=0:H=0:V=0:H1=0:V1=
0
1030 GOSUB 29000:GOSUB 27000:GOSUB 23000
1050 GOSUB 24000:SX=10:SY=11:COLOR 164:PLOT SX,SY
1999 REM * Hauptprogramm *
2000 S(15)=S(ST):ST=STICK(0):IF STRIG(0)=0 THEN H1=H:V1=V:GOSUB 25000
2005 IF ST=15 THEN KF=KF+0.3:FF=FF+0.3:GOTO 2050
2010 H=-(ST>8 AND ST<12)*(SX>0)+(ST<8)*(SX<19)
2020 V=-(ST=14)*(SY>1)+(ST=13)*(SY<22)
2030 KF=KF+1:FF=FF+1:LOCATE SX+H,SY+V,Z:IF Z<>32 THEN COLOR S(ST):PLOT SX,SY:GOT
O 2050
2040 COLOR 32:PLOT SX,SY:SX=SX+H:SY=SY+V:COLOR S(ST):PLOT SX,SY
2050 IF KF>50-SG THEN KF=0:GOSUB 23000
2060 GOSUB 28000:GOTO 2000
15999 REM * Spielanleitung *
16000 GRAPHICS 17:POKE 708,198:POSITION 1,11:? #6;"SPIELREGELN? (J/N)":POKE 764,
255
16010 A=PEEK(764):IF A=255 THEN 16010
16020 IF A=35 THEN RETURN
16030 IF A<>1 THEN 16010
16040 GRAPHICS 0:POKE 709,0:POKE 710,202:POKE 712,198:POKE 82,0:POKE 752,1:POKE
756,204
16050 ? "üDu musst versuchen, durch abschiessen   der roten Sheriffs so viele Pu
nkte wie"
16060 ? "mglich zu erreichen. Dabei wirst du vonkleinen, gr
nen Mnnchen behindert (kei-";
16070 ? "ne Marsmenschen!), die dir nachstellen  und dich vernichten wollen. Die
 Stadt in";
16080 ? "der Mitte des Spielfeldes gewhrt dir   Schutz vor diesen Feinden. Wenn
alle"
16090 ? "Sheriffs abgeschossen sind, gibt es ei- nen Bonus und eine neues Feld w
ird auf-"
16100 ? "gebaut.":? "Es gibt 9 Schwierigkeitsgrade, wobei 1  einfach und 9 schwe
r bedeutet."
16110 POSITION 0,15:? "      *** Bitte eine Taste dr
cken ***";:POKE 764,255
16120 IF PEEK(764)=255 THEN 16120
16130 RETURN
16999 REM * Sound bei Feindbewegung *
17000 SOUND 0,0,0,6:SOUND 0,0,0,0:RETURN
17999 REM * Alle Sheriffs getroffen *
18000 SG=SG+1*(SG<10):SC=SC+100
18010 ? #6;"ü":RESTORE 18500:B=8
18020 FOR A=2 TO 6:POSITION A,B:READ C:? #6;CHR$(C):B=B+1
18030 C=INT(RND(0)*50)+50:FOR D=10 TO 0 STEP -0.5:SOUND 0,C,10,D:NEXT D:SOUND 0,
0,0,0
18040 NEXT A:POSITION 8,10:? #6;"100 Points"
18050 COLOR 42:PLOT 0,6:DRAWTO 19,6:DRAWTO 19,14:DRAWTO 0,14:DRAWTO 0,6
18060 POSITION 4,17:? #6;"PrUss  stOrt":POSITION 4,19:? #6;"t_  S_^tY^uU"
18070 IF PEEK(53279)<>6 THEN 18070
18080 GOTO 1000
```

```
18500 DATA 98,111,110,117,115
18999 REM * Schuss loeschen *
19000 COLOR 32:PLOT PX-H1,PY-V1:COLOR S(ST):PLOT SX,SY:RETURN
19999 REM * Sound bei Treffer *
20000 FOR A=0 TO 40 STEP 2:SOUND 0,A,0,8:NEXT A:SOUND 0,0,0,0:RETURN
20999 REM * Alle Leben verloren *
21000 COLOR 32:PLOT FX(FN),FY(FN):COLOR 133:PLOT FX(FN)+FX,FY(FN)+FY
21010 POSITION 6,11:? #6;"game over":RESTORE 21500
21020 READ A,B:IF A=-1 THEN SOUND 0,0,0,0:GOTO 21040
21030 SOUND 0,A,10,8:FOR C=1 TO B:NEXT C:GOTO 21020
21040 POSITION 5,13:? #6;"Press start"
21050 IF PEEK(53279)<>6 THEN 21050
21060 GOTO 200
21500 DATA 96,60,81,190,72,60,81,60,91,90,68,190,0,60
21510 DATA 96,60,81,190,72,60,81,60,91,90,108,190
21520 DATA -1,-1,-1
21999 REM * Ein Leben verloren *
22000 LE=LE-1:IF LE=0 THEN GOSUB 24000:GOTO 21000
22010 FOR A=1 TO AF:COLOR 32:PLOT FX(A),FY(A):NEXT A:AF=0:FN=0:FF=0:KF=0
22020 PLOT SX,SY:FOR A=100 TO 150:SOUND 0,A,10,10:NEXT A:SOUND 0,0,0,0
22030 FOR A=10 TO 0 STEP -0.2:SOUND 0,100,10,A:NEXT A:SOUND 0,0,0,0:GOTO 1050
22999 REM * Kaktusberechnungen *
23000 COLOR 32:PLOT KX(KN),KY(KN):POKE 77,0
23010 KN=KN+1:IF KN>AK THEN KN=1
23020 COLOR 135:PLOT KX(KN),KY(KN):RETURN
23999 REM * Anzeige *
24000 POSITION 0,0:? #6;"SCORE : ";SC:POSITION 0,23:? #6;"SCHOONER : ";LE:RETURN
24999 REM * Schuss *
25000 SO=0:PX=SX:PY=SY:TRAP 25300
25010 SO=SO+1:PX=PX+H1:PY=PY+V1:IF PY<1 OR PY>22 THEN 25100
25020 LOCATE PX,PY,Z:IF Z<>32 THEN SOUND 0,0,0,0:GOTO 25100
25030 COLOR 32:PLOT PX-H1*(PX-H1<>SX),PY-V1*(PY-V1<>SY):COLOR 169:PLOT PX,PY
25040 SOUND 0,SO,0,6:KF=KF+0.1:FF=FF+0.1:GOTO 25010
25100 IF Z=7 THEN 25300
25110 IF Z=38 THEN AS=AS-1:SC=SC+5:COLOR 32:PLOT PX,PY:GOSUB 19000:GOSUB 24000:G
OSUB 20000:GOTO 25290
25120 IF Z=133 THEN 25500
25290 IF AS=0 THEN POP :GOTO 18000
25300 SOUND 0,0,0,0:GOSUB 19000:PLOT SX,SY:RETURN
25499 REM * Feind getroffen *
25500 COLOR 7:PLOT PX,PY:SC=SC+50:GOSUB 24000:GOSUB 19000:AK=AK+1:IF AK>150 THEN
 AK=1
25510 GOSUB 20000:KX(AK)=PX:KY(AK)=PY
25520 FOR A=1 TO AF:IF FX(A)=PX AND FY(A)=PY THEN 25530
25525 NEXT A:GOTO 25300
25530 POP :FOR B=A TO AF-1:FX(B)=FX(B+1):FY(B)=FY(B+1):NEXT B:AF=AF-1:FN=0:GOTO
25300
25999 REM * Neuer Feind erscheint *
26000 IF AF=6 THEN RETURN
26010 AF=AF+1:FX(AF)=KX(KN):FY(AF)=KY(KN):TRAP 26500
26020 FX=SGN(SX-FX(AF)):FY=SGN(SY-FY(AF)):LOCATE FX(AF)+FX,FY(AF)+FY,Z:IF Z<>32
THEN 26500
26030 FX(AF)=FX(AF)+FX:FY(AF)=FY(AF)+FY:COLOR 133:PLOT FX(AF),FY(AF):TRAP 40000:
RETURN
26500 AF=AF-1*(AF>1):TRAP 40000:RETURN
26999 REM * Aufbau der Sheriffs *
27000 COLOR 38:FOR A=1 TO 30+SG
27010 B=INT(RND(0)*20):C=INT(RND(0)*22)+1:IF B>6 AND B<14 AND C>8 AND C<16 THEN
27010
```

```
27020 LOCATE B,C,Z:IF Z=32 THEN PLOT B,C:NEXT A:AS=30+SG:RETURN
27030 GOTO 27010
27999 REM * Bewegung der Feinde *
28000 IF FF>20-SG OR AF=0 THEN FF=0:GOSUB 26000
28010 FN=FN+1:IF FN>AF OR FN<1 THEN FN=1
28020 FX=SGN(SX-FX(FN)):FY=SGN(SY-FY(FN))
28030 IF FX(FN)+FX>7 AND FX(FN)+FX<13 AND FY(FN)+FY>9 AND FY(FN)+FY<15 THEN RETU
RN
28040 LOCATE FX(FN)+FX,FY(FN)+FY,Z
28050 IF Z=32 THEN COLOR 32:PLOT FX(FN),FY(FN):FX(FN)=FX(FN)+FX:FY(FN)=FY(FN)+FY
:GOSUB 17000:GOTO 28080
28060 IF Z>160 AND Z<165 THEN POP :GOTO 22000
28080 COLOR 133:PLOT FX(FN),FY(FN):RETURN
28999 REM * Aufbau der Kakteen *
29000 COLOR 7:FOR A=1 TO 15+SG
29010 B=INT(RND(0)*20):C=INT(RND(0)*22)+1:IF B>6 AND B<14 AND C>8 AND C<16 THEN
29010
29020 LOCATE B,C,Z:IF Z=32 THEN PLOT B,C:KX(A)=B:KY(A)=C:NEXT A:AK=15+SG:RETURN
29030 GOTO 29010
29999 REM * Schwierigkeitsgrad *
30000 GRAPHICS 17:POKE 708,150:POKE 709,198:POKE 710,168
30010 POSITION 5,11:? #6;"LEVEL : 1":SG=1
30020 POSITION 3,4:? #6;"tombstone city"
30030 POSITION 4,17:? #6;"prUss stOrt":POSITION 5,19:? #6;"t_  RUWY^":FOR A=1 TO
 100:NEXT A
30040 IF PEEK(53279)=6 THEN RETURN
30050 IF PEEK(53279)<>5 THEN 30040
30060 SG=SG+1:IF SG>9 THEN SG=1
30070 POSITION 13,11:? #6;CHR$(SG+48)
30080 FOR A=12 TO 0 STEP -0.5:SOUND 0,20,10,A:SOUND 0,22,10,A:NEXT A
30090 SOUND 0,0,0,0:GOTO 30040
30999 REM * Aufbau der Stadt *
31000 FOR A=10 TO 14 STEP 2:POSITION 8,A:? #6;"ä ä ä":NEXT A:RETURN
31999 REM * Zeichen definieren *
32000 RESTORE 32050:FOR A=1 TO 44:READ B:A$(A,A)=CHR$(B):SOUND 0,B,10,6:NEXT A
32010 SOUND 0,0,0,0:X=USR(ADR(A$),57344,16384,1024)
32020 READ C:IF C=-1 THEN SOUND 0,0,0,0:POKE 559,34:RETURN
32030 FOR A=0 TO 7:READ B:SOUND 0,B,10,6:POKE 16384+C*8+A,B:NEXT A:GOTO 32020
32050 DATA 104,201,3,208,254,162,6,104,149,211,202,208,250,230,212,230,213
32051 DATA 198,212,208,4,198,213,240,18,161,216,129,214,230,216,208,2,230
32052 DATA 217,230,214,208,234,230,215,208,230,96
32100 DATA 1,24,24,24,24,60,126,126,126
32101 DATA 2,126,126,126,60,24,24,24,24
32102 DATA 3,96,112,126,126,126,112,96,0
32103 DATA 4,6,14,126,126,126,14,6,0
32104 DATA 5,60,126,126,126,60,36,66,66
32105 DATA 6,0,0,84,56,124,56,84,0
32106 DATA 7,16,20,92,80,112,16,16,24
32107 DATA 8,0,126,126,126,126,126,126,0
32108 DATA 9,0,0,24,24,24,0,0,0
32767 DATA -1
```

Schneevogel

Schneevogel
für Atari 600/800 XL
Mit dem Joystick kannst Du in diesem Spiel den ‚Schneevogel' fliegen lassen. Bei eingedrücktem Feuerkopf kannst Du ihn so bewegen, daß er in der Lage ist, Mäuse zu fangen. Versuche also, den Vogel über die Mäuse zu bringen und sie zu berühren. Ist die gefangene Maus von brauner Farbe gewesen, erhältst Du 100 Punkte. Willst Du jedoch noch mehr Punkte sammeln, dann paß auf: für kurze Zeit verfärbt sich die braune Maus in eine weiße. Wenn Du jetzt schnell genug reagierst und die Maus fängst, gibt es 1000 Punkte.

Bei jeder gefangenen Maus bricht aber auch ein kleines Stück Erde auf, wo die Maus herauskommen kann. Nach ungefähr 20 gefangenen Mäusen kommt ein Rasenmäher und deckt alle Löcher wieder ab.

Das Programm ist in Basic geschrieben und benötigt ungefähr 7-8 Ram. Nach dem Eingeben ‚Start' drücken und warten. Dann die Labyrinth-Zahl angeben und ‚Return' drücken. Viel Spaß!



```
10 REM ****SCHNEEVOGEL*****
20 REM * BY RICHARD KAEFER *
30 REM * GOETZENTURMSTR.37 *
40 REM *  7100 HEILBRONN    *
50 REM ********************
100 GOSUB 4055:GOTO 500
400 REM FLY BIRD
410 POKE 77,0:LOOK=PEEK(BIRD-20)
415 IF LOOK=63 THEN 450
420 IF LOOK<>0 OR BIRD-20<SC THEN RETURN
430 POKE BIRD,0:POKE BIRD-20,58+64:BIRD=BIRD-20
440 IF STRIG(0)=0 THEN 400
449 RETURN
450 POKE BIRD,0:POKE BIRD-20,58+64:POKE BIRD-40,58+64:POKE BIRD-20,63:BIRD=BIRD-
40
459 RETURN
500 ST=STICK(0):POKE BIRD,58+64
510 IF PEEK(BIRD+20)=0 THEN POKE BIRD,0:BIRD=BIRD+20:POKE BIRD,58+64:POKE 53760,
50:POKE 53761,8
515 POKE 53760,0:POKE 53761,0
520 IF STRIG(0)=0 THEN GOSUB 400
535 NBIRD=BIRD+(ST=7)-(ST=11)
540 IF NBIRD<SC OR NBIRD>SC+479 THEN 500
550 IF PEEK(NBIRD)=0 THEN POKE BIRD,0:BIRD=NBIRD
551 LOOK=PEEK(BIRD+20)
552 IF LOOK=62 THEN GOTO 850
555 IF LOOK=63 THEN POKE BIRD,0
560 IF LOOK<>187 THEN 580
565 POKE 53761,168:FOR A=250 TO 202 STEP -4:POKE 53760,A:NEXT A
570 POKE BIRD,0:BIRD=BIRD+20:RA=0:SCORE=SCORE+100:HIT=HIT+1:GOTO 590
580 IF LOOK<>123 THEN 590
582 POKE 53761,170:FOR A=150 TO 250 STEP 4:POKE 53760,A:NEXT A
585 POKE BIRD,0:BIRD=BIRD+20:RA=0:SCORE=SCORE+1100:HIT=HIT+1
590 IF RND(0)*10>HARD THEN GOSUB 600
595 IF HARD<6 THEN IF HIT=15 THEN POKE MOUSE,0:GOSUB 800
598 POSITION 9,0:? #6;SCORE
599 POKE 53760,0:POKE 53761,0:GOTO 500
600 REM MOVE RATS
```

```
605 IF RA=1 THEN 620
610 RAT=SC+INT(RND(0)*439)+20:IF PEEK(RAT)<>0 THEN 610
612 IF RAT<SC+141 OR RAT>SC+378 THEN 610
615 RA=1:POKE RAT,59+128:POKE RAT+1,63:HOLE=HOLE+1
618 IF HOLE>50 THEN POP :GOTO 4000
620 IF RAT-1<SC+121 THEN POKE RAT,0:RA=0:GOTO 610
625 SEE=PEEK(RAT-1):IF SEE=0 THEN POKE RAT,0:RAT=RAT-1:POKE RAT,59+128
630 IF SEE=63 THEN POKE RAT,0:RA=0:GOSUB 650:RETURN
640 IF RND(0)*10>9.5 THEN POKE RAT,59+64
649 RETURN
650 POKE 53761,136:FOR A=26 TO 116 STEP 10:POKE 53760,A:NEXT A:POKE 53760,0:POKE
 53761,0:RETURN
800 REM MOVE TRACTOR AND FILL HOLES
810 FOR I=SC+120 TO SC+379
815 SOUND 0,100,10,8:SOUND 1,150,8,8
820 POKE I,57+128:FOR A=1 TO 25:NEXT A:POKE I,0:? #6;"      ":REM 6 SPACES
825 GRAPHICS 0:POKE 710,2:POKE 712,3:POKE 752,1:POSITION 9,0:? #6;SCORE:POSITION
 9,0:NEXT I:HOLE=0
830 HARD=HARD-1:IF HARD<4 THEN HARD=4
849 SOUND 0,0,0,0:SOUND 1,0,0,0:HIT=0:RETURN
850 REM LOSE BIRD AND START OVER
855 POKE BIRD,0:FOR A=75 TO 100 STEP 5:SOUND 0,A,10,8:FOR W=1 TO 20:NEXT W:NEXT
A:SOUND 0,0,0,0
860 POKE 53248,0:GOSUB 2050:GOTO 500
1000 REM ENTER LEVEL
1001 GRAPHICS 12+16:POKE 752,1
1002 SETCOLOR 4,5,12:SETCOLOR 0,5,12:POKE 710,6
1003 POSITION 10,5:? #6;"***SCHNEEVOGEL***":FOR X=0 TO 100:NEXT X
1004 POSITION 14,6:? #6;"(c) 1984"
1005 POSITION 10,7:? #6;"by RICHARD KAFFER"
1006 POSITION 0,8:? #6;"0000000000000000000000000000000000000000"
1007 FOR X=100 TO 0 STEP -0.1
1008 SETCOLOR 0,X,9
1009 NEXT X
1010 GRAPHICS 0:POKE 710,68:POKE 712,68:POKE 752,1:POSITION 0,10:? "IN WELCHEM L
ABYRINTH WOLLEN SIE STARTEN?"
1015 TRAP 1015:POKE 752,0:POSITION 13,15:? "LABYRINTH 1 - 6 ";:INPUT LEVEL
1025 IF LEVEL<1 OR LEVEL>6 THEN 1010
1030 HARD=10-LEVEL
1050 GRAPHICS 0:POKE 752,1:POKE 710,N:POSITION 13,5:? "BITTE WARTEN..."
2000 RAMTOP=PEEK(106)-8:POKE 106,RAMTOP:CHBAS=RAMTOP:START=CHBAS*256
2005 FOR I=0 TO 1023:POKE START+I,PEEK(57344+I):NEXT I
2010 FOR I=57 TO 63:MOVE=START+(I*8)
2015 FOR A=0 TO 7:READ SHAPE:POKE (MOVE+A),SHAPE:NEXT A:NEXT I
2050 GRAPHICS 1+16:POKE 756,CHBAS:POKE 712,8:POKE 710,0:POKE 709,255:POKE 710,34
:POKE 708,132
2055 SC=PEEK(88)+256*PEEK(89):BIRD=SC+45
2060 SCORE=0:HOLE=0:HIT=0
2085 GOTO 3000
2089 DATA 0,0,24,24,24,126,36,126
2090 DATA 0,195,36,24,24,36,0,0
2091 DATA 144,112,113,190,0,0,0,0
2092 DATA 100,255,100,255,100,255,100,255
2093 DATA 255,255,255,255,0,0,0,0
2094 DATA 192,240,252,255,76,24,1,75
2095 DATA 13,8,42,168,42,170,170,168,40
2099 DATA 126,126,255,127,127,254,254,255,255,126,126,154,255,254,126,126,255,25
5,255,255,126,127,127,126
3000 FOR I=SC+460 TO SC+479:POKE I,60+192:NEXT I
3005 FOR I=SC+61 TO SC+65:POKE I,61+192:NEXT I:FOR I=SC+101 TO SC+105:POKE I,61+
```

```
192:NEXT I
3015 FOR I=SC+400 TO SC+459:POKE I,62:NEXT I
3020 FOR I=SC TO SC+7:POKE I,60+192:NEXT I:FOR I=SC+13 TO SC+19:POKE I,60+192:NE
XT I
3022 FOR I=SC+380 TO SC+386:POKE I,60+192:NEXT I:FOR I=SC+394 TO SC+399:POKE I,6
0+192:NEXT I
3050 REM P/M GRAPHICS
3055 PM=PEEK(106)-16:POKE 54279,PM:PMBASE=PM*256:POKE 53277,3:POKE 704,244
3060 FOR I=PMBASE+1024 TO PMBASE+1279:POKE I,0:NEXT I:RESTORE 2099
3065 FOR I=PMBASE+1024+50 TO PMBASE+1047+50:READ PMG:POKE I,PMG:NEXT I:FOR I=PMB
ASE+1048+50 TO PMBASE+1250
3070 POKE I,255:NEXT I
3095 POKE 53248,72:POKE 623,1:POKE 559,62
3099 RETURN
4000 REM DRAW WATER
4007 FOR I=SC+385 TO SC+25 STEP -20
4008 SOUND 0,13,8,15
4010 FOR A=-INT(RND(1)*4) TO INT(RND(1)*18):POKE I+A,62:NEXT A
4015 FOR A=1 TO 20:NEXT A:SOUND 0,0,0,0:NEXT I:POKE 712,134:FOR A=1 TO 500:NEXT
A
4025 POKE 712,148:FOR A=1 TO 500:NEXT A:POKE 53248,0:POKE 710,134:POKE 711,134:P
OKE 709,134:POKE 708,134
4030 POKE 712,134:FOR A=1 TO 500:NEXT A
4055 GRAPHICS 0:SETCOLOR 2,4,11:SETCOLOR 1,0,0:SETCOLOR 4,3,4:POKE 756,204:POKE
82,0:POKE 752,1
4060 ? "Aufgabe ist es RATTEN mit einem VOGEL zu fangen."
4070 ? "Dabei muss man,um fliegen zu  koennen,die Feuertaste am Steuerknueppel d
ruecken."
4080 ? "Den Vogel kann man links und rechts durch den Steuerknueppel bewegen."
4090 ? "Um eine Ratte zu fangen muss man von oben drauffliegen."
4100 ? "Eine braune Ratte bringt 100 Punkte,eine weisse dagegen bringt 1000 Punk
te."
4120 ? "Die Punktzahl ist in der oberen Haelfte des Bildschirmes,und wenn man de
n unteren Teil"
4130 ? "mit dem Vogel beruehrt so ist das Spiel aus und man faengt sofort von ne
uem an."
4135 POSITION 5,19:? "PRESS start TO BEGIN THE GAME"
4140 IF PEEK(53279)<>6 THEN 4140
4190 GOTO 1000:GOSUB 2050:GOTO 500
```



Kleinanzeigen

Angebote:

VC-64
Biete echtes Monopoly für VC-64. Wie im Original.
Diskette DM 20,—
Cassette DM 18,—
Chiffre 0401

Commodore 64
irre Programme (375)
zu irren Preisen!
Ausführliche Liste
(48 Seiten) gegen DM 3,-
in Briefmarken.
Detlef Treichel
Mittelbruchzeile 105
1000 Berlin 51

100 neue Programme
für MZ 80 K, MZ 80 A,
MZ 700, VZ 200,
Laser 210, TI 99.

Info, 1,— DM in Marken,
von H. Weisel
Königsberger Str. 20
5412 Ransbach

Kontakte:
Comp.-Club Bad Hersfeld
sucht Mitglieder!
Kontakt über
Roland Reyer
Am Giegenberg 21
6438 Ludwigsau 1
oder Jugendhaus Hersfeld

ZX Spectrum
Soft- und Hardware.
Preiswert
durch Direktimport.

Info bei:
Ursula Kunz
Junge Hälden 3
7500 Karlsruhe 41

Verkaufe
Anrufbeantworter,
Drahtlostelefon,
Telefon - Anlage
Eurosignal 1200.-DM
Div. USA-Telefone
Info bei:
Reinhard Winner
Höchbergerstr. 62
87 Würzburg

COMPUTER-CASSETTEN
im 10er Pack, BASF-Band-LHD,
mit Box, Etikett und Einleger
C 10 nur 15,— DM
C 20 16,— DM; C 30 17,— DM
Brandneu von TDK:
PC-10 im 10er Pack 29,— DM
CASSETTEN-AUFKLEBER
auf Lochstreifen 100 St. 5,— DM
120 St. auf A4-Druckbögen
7,— DM; Versand ab 20,— DM:
Christomenia-Cassettenstudio
Postfach 20
3584 Zwesten, Tel. 0 56 26-2 81

ZX-81:
Großes Softwareangebot.
Spiele, Infos, Tips,
Amateurfunk-Software.

Ausführl. Liste gegen
Rückporto von:
Michael Schramm
Freiligrathstr. 5
2300 Kiel 1

Universelles
Kopierprogramm
Copy für den **ZX Spectrum**
auf Cassette + Catalog –
Programm + Beschreibung
nur DM 16,—

100 Programme für VC-64
wegen Hobby - Aufgabe
zu verkaufen!
Rolf Freitag
Gneisenaustr. 87
4600 Dortmund 1
Tel: 0231 - 825826

Verkaufe
Atari, und jede Menge
Kassetten, z. B.:
Phoenix, Dig-Dug und
Standard Steuerknüppel

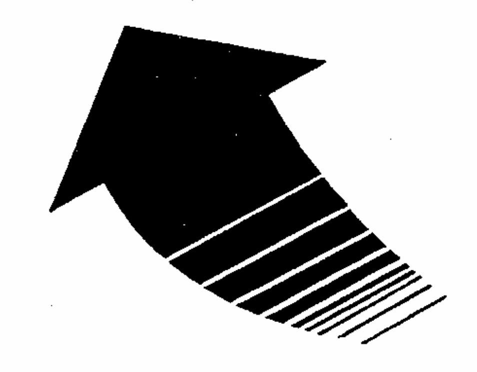
Tel.: 0 79 45-5 61

Tausche **Spectrum-Software,**
Jan Weigner
Carl-Schurz-Straße 11
2800 Bremen
Sinclair forever!!!

A C H T U N G
für TI 99/4A jetzt lieferbar
EXTENDED BASIC DM 295,-
Editor Assembler DM 189,-
Parsec DM 79,-
TI-Logo II DM 320,-
32K-Erw. extern DM 428,-
Modul-Spiele direkt aus den
*** * * * USA * * * * * * ***
zb. Mash, Demon, Baseball,
Tunnels of Doom, Moon mine,
Jawbreaker, Hopper, Attack,
Bigfoot . . . . . . DM 99,-
diverse ATARI-Spiele für TI
Pac Man, Donkey Kong, Dig
Dug, Mrs. Pac Man, Defender,
Centipede . . . DM 99,-
Neu!!! MBX-Sprachsysteme
nur für den TI-99/4A
Spiele direkt mit der Sprache steuern DM 298,-
Neu !!! GRAFIK TABLEAU
für TI 99/4A DM 298,-
für ATARI DM 248,-
für VC 64 DM 298,-
Bitte Preisliste anfordern:
RADIX Bürotechnik
2000 Hamburg 13
Bornstr. 4
Tel. 040/44 16 95 10-18.00 h

Aktion: Billige Kleinanzeigen
Lieber Leser,
wollten Sie eine Kleinanzeige aufgeben? Suchen Sie einen gebrauchten Home-Computer, Software oder Kontakte zu Computer-Clubs etc.? **Dann nutzen Sie jetzt den Kleinanzeigenservice von ‚Computronic'.** ‚Computronic' ist die Zeitschrift für Home-Computer-Besitzer. Sie erreichen durch uns eine Vielzahl von Lesern im In- und Ausland.
Und so wird's gemacht:
Kleinanzeigen-Coupon ausfüllen (alle Zeilen können ausgefüllt werden) und gegen **Vorkasse von DM 4,50 einsenden an:**
Tronic-Verlag
Postfach 41
3444 Wehretal 1

Eingesandt werden muß bis spätestens **1. Juni 1984** (es gilt der Poststempel).
Also, schnell ausfüllen und einwerfen in den nächsten Postkasten. Dann ist mit Sicherheit Ihre Kleinanzeige dabei.





Korrekturen Heft 5

Leider wurden in unserer Ausgabe Mai'84 nicht die Original - Listings der Atari-Spiele » Tank - Battle « und » Oil - Panic « abgedruckt. Wir bitten dies bei allen Atari - Fans zu entschuldigen.

Richtig: Tank - Battle

```
160 SETCOLOR 1,1,11:SETCOLOR 0,11,12:SETCOLOR 2,8,8:SETCOLOR 3,5,7:T=10:T1=10
340 POSITION 0,22:PRINT #6;"TANKS";T;" TANGS:";T1;" ":POSITION 0,20:PRINT #6;"MO
VES:";PG
365 IF S=0 THEN X=19
620 IF P=4 THEN C=0
850 TRAP 940
960 IF K<>189 AND K<>159 AND K<>5 AND K<>166 AND K<>11 AND K<>171 THEN COLOR 32:
RETURN
1110 POKE 559,0:RESTORE 1130
1670 RESTORE 1680:FOR A=1 TO 11:READ CH:PRINT #6;CHR$(CH);:GOSUB 1720:NEXT A
```

Richtig: Oil - Panic

```
230 DIM A$(28):RESTORE 250
410 IF ST<12 AND ST>8 THEN X=X-4:SOUND 0,X+57,10,6
440 POKE 53248,X:SOUND 0,0,0,0
450 IF S=0 THEN W=INT(GW):T=INT(RND(0)*19):S=1:COLOR 134:PLOT T,2:GOTO 390
490 FOR A=240 TO 80 STEP -20:SOUND 1,A,10,8:NEXT A:SOUND 1,0,0,0
500 POKE 53278,0:FOR A=2 TO 21:COLOR 129:PLOT T,A:FOR B=1 TO 10:NEXT B:COLOR 164
:PLOT T,A:NEXT A:S=0
580 FOR A=PM+1024+Y TO PM+1024+Y-H STEP -1:POKE A,255:NEXT A:RETURN
590 FOR A=PM+1024+Y-7 TO PM+1023+Y-H:POKE A,0:NEXT A:RETURN
620 I=0:H=1:GOSUB 550:GOSUB 590:RETURN
630 FOR A=0 TO 20 STEP 2:SOUND 1,A,0,10:SOUND 2,A+10,0,10:NEXT A:SOUND 1,0,0,0:S
OUND 2,0,0,0
740 GOSUB 550:H=H+1:GOSUB 580:RETURN
750 GRAPHICS 17:SETCOLOR 2,14,10:SETCOLOR 1,13,4:SETCOLOR 3,15,5:SETCOLOR 0,10,1
0:POKE 53248,0
780 GOSUB 840:POSITION 0,4:PRINT #6;"PRESENTS :";:POSITION 5,9:GOSUB 840
790 RESTORE 800:FOR A=1 TO 9:READ CH:PRINT #6;CHR$(CH);:GOSUB 840:NEXT A
```

Computronic Bestellkarte-Cassettenservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service, zum Selbstkostenpreis, über den Verlag bezogen werden.

Die Zustellung erfolgt: gegen **Vorkasse** ☐ / per **Nachnahme** ☐ + Versandkosten ☐ innerhalb von 14 Tagen!

Achtung: Disketten können vorerst nur für: Apple / VC-64 / VC-20 bezogen werden!

Entnehmen Sie bitte aus unserer Preisliste Seite 80 die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir: ☐ Cassette für ______ (System) ☐ Anzahl

☐ Diskette für ______ (System) ☐ Anzahl

zum Preis von gesamt: ______ DM

Name: ______ Straße, Nr.: ______

PLZ/Ort: ______ Unterschrift, Datum: ______

Service

Jedes Programm in Computronic wird einer Endkontrolle in unserem Hause unterzogen und genauestens geprüft. Alle im Heft abgedruckten Programme sind nach der richtigen Eingabe der Listings auch funktionsfähig. Viele Leser verlieren jedoch schnell die Geduld am Programmieren, sollte etwas einmal nicht klappen. Die häufigste Ursache von Störungen im Programm, sind unterlaufene Fehler bei der Eingabe. Verzweifeln Sie nicht, sollten Sie einmal keine Zeit zum Programmieren haben oder sollte etwas nicht gelingen. Alle Programme werden im Verlag gespeichert und können jederzeit mit beiliegender Bestellkarte zum Selbstkostenpreis bezogen werden.

Tragen Sie bitte alle notwendigen Angaben in die Bestellkarte ein, und senden sie an:

Tronic-Verlag GmbH, Postfach 41, 3444 Wehretal 1

Preisliste Heft 4 (K) (D)

VC-64 K = 8,– DM, D = 15,– DM
Mauern, Widerstand

ZX-Spectrum
Inventur K = 12,– DM

TI 99 K = 8,50 DM
Drei-Kronen-Spiel
Zahlenputzen

VC-20 K = 11,50 DM, D = 18,– DM
Hangman, Saurer Regen,
Quadr. Gleichung

ZX-81 K = 8,50 DM
Invader, Gun-Man
Landeanflug Boing K = 15,– DM

Dragon 32 K = 6,– DM
Blizzard

Apple II K = 14,50 DM, D = 19,50 DM
Wilder Westen,
Karambolage,
Maskengenerator

Atari K = 10,50 DM
Mastermind,
Schlangenkrieg

Preisliste Heft 5 (K) (D)

TI 99 K = 14,50 DM
Karl der Käfer
Alien-Landing

VC-64 K = 15,50 DM, D = 23,50 DM
Space-Comets
Erdspalte
Sprite-Data

Apple II K = 9,50 DM, D = 17,50 DM
Musik-Maker
Mission-Adler
Disk-Catalog

Sharp MZ 700 Kalender K = 8,50 DM
Sharp PC 1500 Lotto

Dragon 32 K = 10,– DM
Space-Flight, Geosoft

ZX-81 K = 10,– DM
Go-Ball, Grand-Prix

ZX Spectrum K = 6,50 DM
Missile-Comment

Atari K = 12,50 DM
Tank-Battle
Oil Panic

Preisliste Heft 6 (K) (D)

System	Preis	Programme
VC-64	K = 16,50 D = 23,50	„Data Generator" „Autostart" „Bestellschein" „Roadpainter"
ZX-81	K = 10,00	„Moon-Crash" „ZX-Draw"
Dragon 32/64	K = 8,50	„Wargames"
ZX-Spectrum	K = 13,50	„Defender" „Lui der Wurm" „Alternativer Zeichensatz"
Apple II	D = 19,50 K = 12,50	„Snake" „Super Datei" „Shape-tables"
TI-99	K = 14,50	„Jack the Digger" „Noah - 2099"
VC-20	K = 8,50 D = 15,00	„Bestellschein" „Glücky"
Atari	K = 12,50	„Tomstone-City" „Schneevogel"

Computronic **Abonnement**

Bitte ausschneiden und senden an: Tronic-Verlag, Postfach 41, 3444 Wehretal 1.

Hiermit bestelle ich die Zeitschrift »Computronic« ab Heft Nr. ______ zum Jahresabonnementpreis (12 Ausgaben) von DM 55,— incl. Versand, Inland und DM 75,— incl. Versand, Ausland.

Name/Vorname: ______________________________

Straße, Nr. ______________________ Plz, Ort: ______________________

Ich wünsche folgende Zahlungsweise:

☐ Bargeldlos durch Bankeinzug: Bankleitzahl: ______________________

Geldinstitut: ______________________ Konto Nr.: ______________________

☐ gegen Rechnung

☐ gegen Vorkasse
(betreffendendes bitte ankreuzen)

____
Datum, Unterschrift

Abonnements-Kündigungen: 6 Wochen vor Ablauf des Jahres-Abonnement.

Computronic **Kleinanzeigen** **nur gegen Vorkasse**

Bitte veröffentlichen Sie ab nächstmöglicher Ausgabe

unter der Rubrik: ☐ Angebote ☐ Kontakte

☐ Gesuche ☐ Verschiedenes

folgenden Text:

Bitte jeweils 35 Buchstaben pro Zeile! Bitte Absender nicht vergessen!

Preis für Kleinanzeigen:

Private Anzeige	DM 4,50
gewerbl. Anzeige	DM 8,00 je Druckzeile
Chiffregebühr	DM 10,00 je Anzeige

Name: ______________________ Straße, Nr.: ______________________

Vorname: ______________________ PLZ/Ort: ______________________

Unterschrift, Datum: ______________________ privat ☐ gewerbl. ☐
(bitte ankreuzen)

DINO EGGS

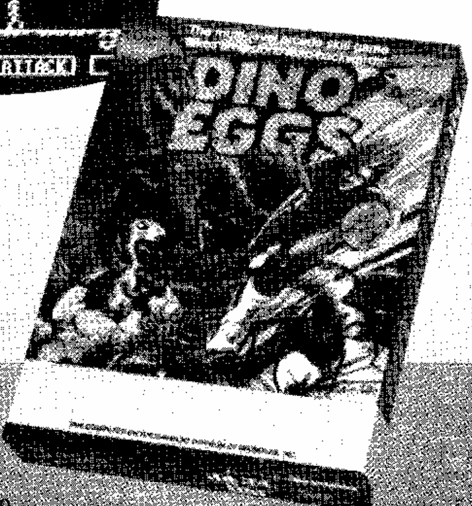
Can you save the Doomed Dinos and escape the Dino Mom's stomp?

Warped into a prehistoric world you've contaminated the dinosaurs. You must climb nine deadly cliffs, find the dino eggs and carry them back to the 21st century.

Dodge the radioactive snakes and spiders when you climb, leap and jump over the challenging cliffs to locate the eggs. As you fight your way up the nine levels your skills must increase if you are to survive the deadly creatures

Fire is the only weapon you have to keep the Dino Mom from putting her foot down and crushing YOU!

Can you escape the Dino Mom's stomp, avoid the crawling, clawing creatures, find the eggs and save the dinosaurs from extinction? Only you can face the challenges of survival in the prehistoric time warp of DINO EGGS™ from MicroFun, Inc. . . . *the fun goes on forever.*



Created by David Schroeder

Available for:
Apple II™ & IIe™
Atari 800™
IBM PC™
ColecoVision™
Commodore 64™

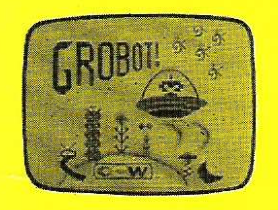
. . . the fun goes on forever

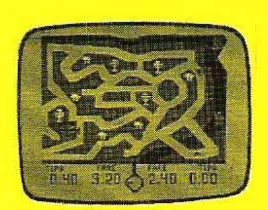
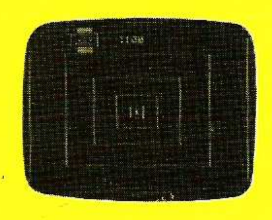
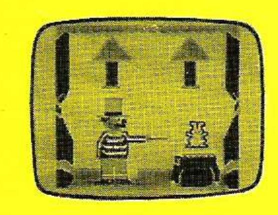
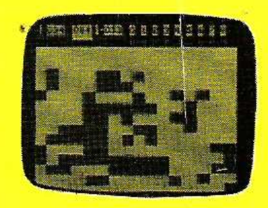
Heft 7 von Computronic ist:

Ab 25. Juni bei Ihrem Zeitschriftenhändler erhältlich!



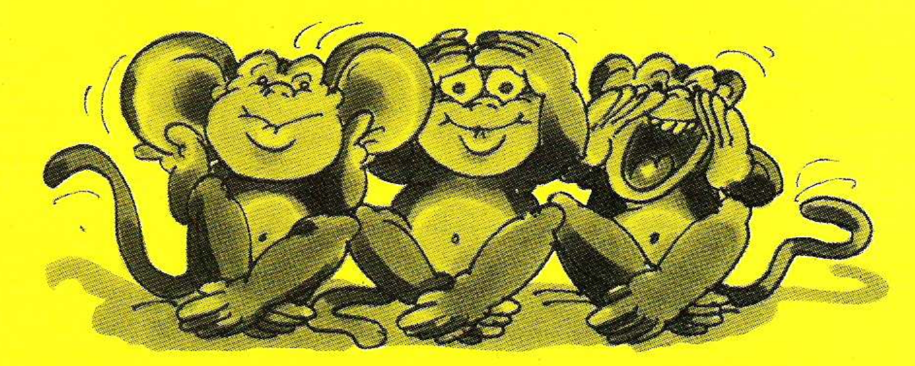
Natürlich bringen wir dann wieder
jede Menge Programme!
Programme!
Programme!



und:

Neues, Infos, Tips + Tricks!



Es verabschiedet sich: Die Computronic-Redaktion

Impressum: **Computronic** erscheint monatlich im Verlag: Tronic-Verlagsgesellschaft mbH, Landstraße 29, 3444 Wehretal 1, Tel.: (05651) 40643/40693. **Redaktion:** Rainer Beck (verantwortlich), H. A. Ederg. **Freie Mitarbeiter:** Frank Brall, G. Wagner. **Satz und Reproduktion:** Intragrafik **Druck:** Schimmel KG, Würzburg. **Vertrieb:** Inland (Groß-, Einzel- und Bahnhofsbuchhandel), sowie Österreich und Schweiz: Verlagsunion, 6200 Wiesbaden. **Anzeigenverwaltung:** E. Herwig. **Anzeigenpreise:** Es gilt die Anzeigenliste Nr. 1. Bitte Media-Unterlagen anfordern. **Bezugspreise:** Einzelheft (Inland) DM 5,50, Abonnement (Inland) DM 55,—, (Ausland) DM 75,—. **Autoren und Manuskripte:** Bei Zusendung von Manuskripten und Datenträger erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger. Für die mit Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung. Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Nachdruck (auch auszugsweise) und Vervielfältigung nur mit ausdrücklicher Genehmigung des Verlages. Für unaufgeforderte Einsendungen von Manuskripten, Tonträgern und Software wird keine Haftung übernommen.

Wir bedanken uns für die freundl. Unterstützung der Fa: »Foto Dunkel« 3440 Eschwege



computronic

Infos

Tips und Tricks für Einsteiger, Anfänger und Fortgeschrittene. Informationen und Daten. Sinnvolle Anwendungen und Anregungen. Lernen, anhand von Computronic, Programme einzugeben und Programme zu schreiben. Von Basic bis Maschinensprache. Neues vom Computermarkt über Hard- und Software.

computronic

Software-Listings

Unser Schwerpunkt in den Ausgaben (70 %). Leichte und anspruchsvolle Programme werden im Heft abgedruckt, wobei die Redaktion alle bekannten Home-Computer berücksichtigen wird. Wenn man will, kann man die Programme einfach eingeben, damit spielen oder arbeiten. Alle Konzepte und Begriffe sowie alle Spiele werden vor ihrer Anwendung erklärt.

computronic

Software-Service

Abgedruckte Programme können als zusätzlicher Service über den Verlag bezogen werden. Wer kein Interesse hat seinen Computer zu »füttern«, oder wem es zu langweilig wird zu programmieren, kann alle abgedruckten interessanten Spiele auf Cassette und Diskette erwerben.

Computronic erscheint monatlich im Tronic-Verlag. Der Bezugspreis beträgt im Inland DM 5,50. Im Preis ist die gesetzliche Mehrwertsteuer enthalten.

Tronic-Verlag GmbH · Postfach 41 · 3444 Wehretal 1 · 05651-40643/40693